METHOD AND SYSTEM FOR INCORPORATING SECURITY MECHANISM INTO SESSION INITIATION PROTOCOL REQUEST MESSAGE FOR CLIENT PROXY AUTHENTICATION

Patent number: JP2003108527 (A)
Publication date: 2003-04-11

Inventor(s): BOBDE NIKHIL P; DEMIRTJIS ANN; HAN MU
Applicant(s): MICROSOFT CORP

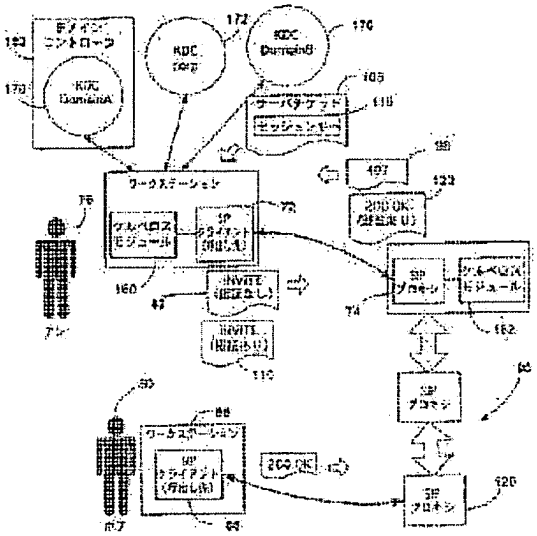
Classification:
- international: ***G06F21/20; G06F15/00; G09C1/00; H04L29/06;*** **G06F21/20; G06F15/00; G09C1/00; H04L29/06;** (IPC1-7): G06F15/00; G09C1/00
- european: H04L29/06S12A; H04L29/06C2; H04L29/06M2H2; H04L29/06S8A
Application number: JP20020174951 20020614
Priority number(s): US20010298239P 20010614; US20020151747 20020517

Also published as:
JP4294268 (B2)
EP1267548 (A2)
EP1267548 (A3)
EP1267548 (B1)
US2008022383 (A1)

more >>

Abstract of **JP 2003108527 (A)**

PROBLEM TO BE SOLVED: To provide a method and a system for allowing an SIP client and an SIP proxy to authenticate each other by incorporating a Cerberus security mechanism into a message flow of signaling operation based on a session initiation protocol. SOLUTION: When receiving a request message such as an INVITE request from the SIP client, the SIP proxy sends a challenge message for indicating the necessity of authentication based on Cerberus in response to this message. The SIP client sends a second request message having a proxy authorization header including authenticating data including a Cerberus server ticket to a proxy in response to this message so that the proxy can authenticate a user of the client.



(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2003－108527
（P2003－108527A）
(43)公開日　平成15年４月11日(2003.4.11)

(51)Int.Cl.⁷	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　15/00	３３０	Ｇ０６Ｆ　15/00	３３０Ｂ	５Ｂ０８５
	３１０		３１０Ｄ	５Ｊ１０４
Ｇ０９Ｃ　1/00	６４０	Ｇ０９Ｃ　1/00	６４０Ｅ	

審査請求　未請求　請求項の数28　ＯＬ　（全 23 頁）

(21)出願番号　特願2002－174951(P2002－174951)

(22)出願日　平成14年６月14日(2002.6.14)

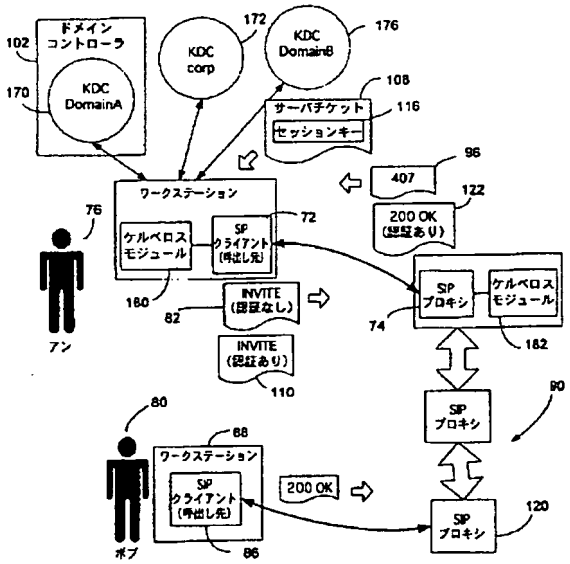
(31)優先権主張番号　６０／２９８，２３９
(32)優先日　平成13年６月14日(2001.6.14)
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　１０／１５１，７４７
(32)優先日　平成14年５月17日(2002.5.17)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　391055933
マイクロソフト　コーポレイション
ＭＩＣＲＯＳＯＦＴ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　ワシントン州　98052－6399　レッドモンド　ワン　マイクロソフト　ウェイ　（番地なし）
(72)発明者　ニキル　ピー．ポブデ
アメリカ合衆国　98052　ワシントン州　ベルビュー　43　プレイス　14405
(74)代理人　100077481
弁理士　谷　義一　（外２名）

最終頁に続く

(54)【発明の名称】　クライアント・プロキシ認証のためにセッションイニシエーションプロトコルリクエストメッセージにセキュリティ機構を組み込むための方法およびシステム

(57)【要約】
【課題】　セッションイニシエーションプロトコルに基づくシグナリングオペレーションのメッセージフローにケルベロスセキュリティ機構を組み込んで、ＳＩＰクライアントとＳＩＰプロキシが相互に認証することを可能にする方法およびシステムを提供する。
【解決手段】　ＩＮＶＩＴＥリクエストなどのリクエストメッセージをＳＩＰクライアントから受け取ると、ＳＩＰプロキシはこれに応答して、ケルベロスに基づく認証が必要であることを指示するチャレンジメッセージを送る。これに応答してＳＩＰクライアントは、プロキシに対するケルベロスサーバチケットを含む認証データを含んだｐｒｏｘｙ　ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有する第２のリクエストメッセージを送って、プロキシがクライアントのユーザを認証できるようにする。



1

【特許請求の範囲】

【請求項１】　セッションイニシエーションプロトコル（ＳＩＰ）クライアントのユーザを認証するために、ＳＩＰプロキシによりステップを実行するためのコンピュータ実行可能命令を有するコンピュータ可読媒体であって、前記ステップが、

前記ＳＩＰクライアントから第１のリクエストメッセージを受け取るステップと、

前記第１のリクエストメッセージが、前記ＳＩＰクライアントのユーザを認証するための認証データを含んでいないと判定するステップと、

認証が必要であることを指示するコードを含むチャレンジメッセージを送るステップと、

選択された認証プロトコルに従って前記ＳＩＰクライアントのユーザを認証するための認証データを含むｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを含む第２のリクエストメッセージを前記ＳＩＰクライアントから受け取るステップと、

前記第２のリクエストメッセージのｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダの中の認証データを使用して前記ＳＩＰクライアントのユーザを認証するステップとを備えることを特徴とするコンピュータ可読媒体。

【請求項２】　請求項１に記載のコンピュータ可読媒体であって、前記第１および第２のリクエストメッセージがＳＩＰ　ＩＮＶＩＴＥリクエストであることを特徴とするコンピュータ可読媒体。

【請求項３】　請求項１に記載のコンピュータ可読媒体であって、前記ＳＩＰクライアントのユーザの認証に成功した後に、前記第２のリクエストメッセージを、前記リクエストメッセージの中で識別される意図された呼出し先に至るＳＩＰシグナリング経路に転送するステップを実行するためのコンピュータ実行可能命令をさらに有することを特徴とするコンピュータ可読媒体。

【請求項４】　請求項１に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがケルベロスプロトコルであり、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダの中の認証データが、前記ＳＩＰプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むことを特徴とするコンピュータ可読媒体。

【請求項５】　請求項４に記載のコンピュータ可読媒体であって、前記認証ステップが、ケルベロスモジュールを呼び出して前記ケルベロスサーバチケットの有効性をチェックするステップと、前記ケルベロスサーバチケットから、前記ＳＩＰクライアントと通信するのに使用するセッションキーを抜き出すステップとを含むことを特徴とするコンピュータ可読媒体。

【請求項６】　請求項１に記載のコンピュータ可読媒体であって、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダの中の認証データが、前記ＳＩＰクライアントと前記ＳＩＰプロキシとの間の相互認証を要求するデータを含み、前記コンピュータ可読媒体が、前記ＳＩＰクライアントが前記ＳＩＰプロキシを認証するのに使用する前記ＳＩＰプロキシの認証データを含むｐｒｏｘｙ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎヘッダを有するメッセージを、前記ＳＩＰクライアントに返すステップを実行するコンピュータ実行可能命令をさらに有することを特徴とするコンピュータ可読媒体。

2

【請求項７】　請求項１に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがＮＴＬＭプロトコルであることを特徴とするコンピュータ可読媒体。

【請求項８】　セッションイニシエーションプロトコル（ＳＩＰ）クライアントが、ＳＩＰプロキシを介したセッションの開始に関連して前記ＳＩＰクライアントのユーザを前記ＳＩＰプロキシに対して認証するステップを実行するためのコンピュータ実行可能命令を有するコンピュータ可読媒体であって、前記ステップが、意図された呼出し先に対する第１のリクエストメッセージを前記ＳＩＰプロキシに送るステップと、

前記第１のリクエストメッセージに対する応答として前記ＳＩＰプロキシから送られた、認証が必要であることを指示するチャレンジメッセージを受け取るステップと、

選択された認証プロトコルに従って前記ユーザを認証するための認証データを含むｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを構築するステップと、前記構築されたｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを含む、前記意図された呼出し先に対する第２のリクエストメッセージを送るステップとを備えることを特徴とするコンピュータ可読媒体。

【請求項９】　請求項８に記載のコンピュータ可読媒体であって、前記第１および第２のリクエストメッセージがＳＩＰ　ＩＮＶＩＴＥリクエストであることを特徴とするコンピュータ可読媒体。

【請求項１０】　請求項８に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがケルベロスプロトコルであり、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダの中の認証データが、前記ＳＩＰプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むことを特徴とするコンピュータ可読媒体。

【請求項１１】　請求項８に記載のコンピュータ可読媒体であって、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを構築するステップが、ケルベロスキーディストリビューションセンターから前記ケルベロスサーバチケットを取得するステップを含むことを特徴とするコンピュータ可読媒体。

【請求項１２】　請求項１１に記載のコンピュータ可読

媒体であって、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダが、前記ＳＩＰクライアントと前記ＳＩＰプロキシとの間の相互認証のリクエストを表すデータを含み、前記コンピュータ可読媒体が、
前記第２のリクエストメッセージに対する応答として前記ＳＩＰプロキシから応答メッセージを受け取るステップと、
前記応答メッセージの中に含まれるｐｒｏｘｙ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ ｉｎｆｏｒｍａｔｉｏｎヘッダから、前記ＳＩＰプロキシの認証データを抜き出すステップと、
前記ｐｒｏｘｙ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ ｉｎｆｏｒｍａｔｉｏｎヘッダから抜き出した前記ＳＩＰプロキシの認証データに基づいて前記ＳＩＰプロキシを認証するステップとを実行するためのコンピュータ実行可能命令をさらに含むことを特徴とするコンピュータ可読媒体。

【請求項１３】　請求項８に記載のコンピュータ可読媒体であって、前記ＳＩＰクライアントが、
前記選択された認証プロトコルに従って前記ＳＩＰクライアントのユーザを認証するためのユーザ認証データを取得するステップと、
前記ＳＩＰプロキシに登録するために、前記ユーザを認証するための認証データを含むｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有するＲＥＧＩＳＴＥＲメッセージを前記ＳＩＰプロキシに送信するステップとを実行するためのコンピュータ実行可能命令をさらに有することを特徴とするコンピュータ可読媒体。

【請求項１４】　請求項１３に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがケルベロスプロトコルであり、前記ユーザの認証データが、ケルベロスキーディストリビューションセンターから取得した、前記ＳＩＰプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むことを特徴とするコンピュータ可読媒体。

【請求項１５】　請求項８に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがＮＴＬＭプロトコルであることを特徴とするコンピュータ可読媒体。

【請求項１６】　セッションイニシエーションプロトコル（ＳＩＰ）プロキシが、セッション開始オペレーションの間にＳＩＰクライアントのユーザを認証するための方法であって、
前記ＳＩＰクライアントから第１のリクエストメッセージを受け取るステップと、
前記第１のリクエストメッセージが、前記ＳＩＰクライアントのユーザを認証するための認証データを含んでいないと判定するステップと、
「４０７　Ｐｒｏｘｙ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｒｅｑｕｉｒｅｄ」ステータスコードを含むメッセージを前記ＳＩＰクライアントに送って、認証が必要であることを指示するステップと、
前記ＳＩＰクライアントのユーザを認証するためのユーザ認証データを含むｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを含む第２のリクエストメッセージを前記ＳＩＰクライアントから受け取るステップであって、前記ユーザ認証データが、前記ＳＩＰプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むステップと、
前記ケルベロスサーバチケットを使用して前記ＳＩＰクライアントのユーザを認証し、前記ＳＩＰクライアントとの通信を暗号化するためのセッションキーを前記ケルベロスサーバチケットから抜き出すステップと、
前記第２のリクエストメッセージを、前記ＩＮＶＩＴＥメッセージの中で識別される意図された呼出し先に至るＳＩＰシグナリング経路に転送するステップとを備えることを特徴とする方法。

【請求項１７】　請求項１６に記載の方法であって、前記第１および第２のリクエストメッセージがＳＩＰ　ＩＮＶＩＴＥリクエストであることを特徴とする方法。

【請求項１８】　請求項１６に記載の方法であって、前記第２のリクエストメッセージの中のｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダの認証データが、前記ＳＩＰクライアントと前記ＳＩＰプロキシとの間の相互認証を要求するデータを含み、前記方法が、前記ＳＩＰクライアントが前記ＳＩＰプロキシを認証するのに使用する認証データを含むｐｒｏｘｙ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ ｉｎｆｏｒｍａｔｉｏｎヘッダを有するメッセージを前記ＳＩＰクライアントに返すステップをさらに備えることを特徴とする方法。

【請求項１９】　セッションイニシエーションプロトコル（ＳＩＰ）クライアントが、ＳＩＰプロキシを介したセッションの開始に関連して前記ＳＩＰクライアントのユーザを前記ＳＩＰプロキシに対して認証するための方法であって、
意図された呼出し先に対する第１のリクエストメッセージを前記ＳＩＰプロキシに送るステップと、
前記第１のリクエストメッセージに対する応答として前記ＳＩＰプロキシにより送られた、認証が必要であることを指示するチャレンジメッセージを受け取るステップと、
前記ユーザを認証するためのユーザ認証データを含むｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを構築するステップであって、前記ユーザ認証データが、前記ＳＩＰプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むステップと、
前記構築されたｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを含む、前記意図された呼出し先に対する第２のリクエストメッセージを送るステップとを備えることを特徴とする方法。

【請求項２０】　請求項１９に記載の方法であって、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを構築するステップが、ケルベロスキーディストリビューションセンターから前記ケルベロスサーバチケットを取得するステップを含むことを特徴とする方法。

【請求項２１】　請求項１９に記載の方法であって、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを構築するステップが、前記ＳＩＰクライアントと前記ＳＩＰプロキシとの間の相互認証のリクエストを、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダに挿入するステップを含み、前記方法が、

前記第２のリクエストメッセージに対する応答として前記ＳＩＰプロキシから応答メッセージを受け取るステップと、

前記応答メッセージの中に含まれるｐｒｏｘｙ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎヘッダから、前記ＳＩＰプロキシの認証データを抜き出すステップと、

前記ｐｒｏｘｙ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎヘッダから抜き出した前記ＳＩＰプロキシの認証データに基づいて前記ＳＩＰプロキシを認証するステップとをさらに備えることを特徴とする方法。

【請求項２２】　請求項２１に記載の方法であって、前記第１および第２のリクエストメッセージがＳＩＰ　ＩＮＶＩＴＥリクエストであることを特徴とする方法。

【請求項２３】　セッションイニシエーションプロトコル（ＳＩＰ）クライアントが、ＳＩＰプロキシと認証を実行するための方法であって、

ケルベロス認証プロトコルに従って前記ＳＩＰクライアントを認証するための認証データを取得するステップであって、前記認証データが、前記ＳＩＰプロキシにアクセスするためのサーバチケットを含むステップと、

前記ＳＩＰプロキシに登録するために、前記認証データを含むｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有するＲＥＧＩＳＴＥＲメッセージを前記ＳＩＰプロキシに送信するステップとを備えることを特徴とする方法。

【請求項２４】　請求項２３に記載の方法であって、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダが、前記ＳＩＰサーバとの相互認証のリクエストを含み、前記方法が、

前記ＲＥＧＩＳＴＥＲメッセージに対する応答として前記ＳＩＰプロキシから応答メッセージを受け取るステップと、

前記応答メッセージの中に含まれるｐｒｏｘｙ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎヘッダから、前記ＳＩＰプロキシの認証データを抜き出すステップと、

前記ｐｒｏｘｙ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎヘッダから抜き出した前記ＳＩＰプロキシの認証データに基づいて前記ＳＩＰプロキシを認証するステップとをさらに備えることを特徴とする方法。

【請求項２５】　セッションイニシエーションプロトコル（ＳＩＰ）リクエストメッセージを表すデータ構造が記憶されたコンピュータ可読媒体であって、

ＳＩＰプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むデータフィールドを有するｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを含む複数のＳＩＰヘッダと、

メッセージ本体とを備えることを特徴とするコンピュータ可読媒体。

【請求項２６】　請求項２５に記載のコンピュータ可読媒体であって、前記ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダが、前記ケルベロスサーバチケットに関連するセッションキーを使用して前記ＳＩＰリクエストメッセージの一部に署名することによって生成された署名を有する第２のデータフィールドを有することを特徴とするコンピュータ可読媒体。

【請求項２７】　請求項２５に記載のコンピュータ可読媒体であって、前記ＳＩＰリクエストメッセージがＳＩＰ　ＩＮＶＩＴＥリクエストであることを特徴とするコンピュータ可読媒体。

【請求項２８】　請求項２５に記載のコンピュータ可読媒体であって、前記ＳＩＰリクエストメッセージがＳＩＰ　ＲＥＧＩＳＴＥＲリクエストであることを特徴とするコンピュータ可読媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は一般に、コンピュータネットワークを介した装置間通信に関し、詳細には、セキュリティ機構、例えばケルベロス認証プロトコルに基づくセキュリティ機構を、セッションイニシエーションプロトコル（ＳＩＰ：ＳｅｓｓｉｏｎＩｎｉｔｉａｔｉｏｎ　Ｐｒｏｔｏｃｏｌ）を通信セッションを確立するためのシグナリングプロトコルとして使用したネットワーク通信に組み込むことに関する。

【０００２】

【従来の技術】セッションイニシエーションプロトコル（ＳＩＰ）は、コンピューティング装置が、自体がコンピュータネットワークを介して通信したいと欲している他の装置の位置を突き止め、その装置との間に通信セッションを確立するための機構を提供するシグナリングプロトコルである。ＳＩＰは汎用性のあるプロトコルであり、多くの異なるシナリオで通信セッションを確立するのに使用されてきた。ＳＩＰは例えば、インターネット会議、電話、プレゼンス（ｐｒｅｓｅｎｃｅ）、イベント通知およびインスタントメッセージングに使用されている。ＳＩＰの重要な強みは、位置独立の単一のアドレ

スの下にいる被呼者（ユーザ）に、たとえこの被呼者が別のコンピュータに移動していたとしてもたどり着くことができる、パーソナルモビリティ（ｐｅｒｓｏｎａｌ　ｍｏｂｉｌｉｔｙ）をサポートしていることである。

【０００３】ＳＩＰに基づくセッション開始オペレーションの共通する１つのモードは「プロキシ（ｐｒｏｘｙ）モード」である。例えば、ＳＩＰクライアント（「呼出し元」）は、意図された受信者（「呼出し先」）を電子メールアドレスに似たアドレスによって識別するＩＮＶＩＴＥメッセージなどのＳＩＰリクエストメッセージを送ることができる。このリクエストメッセージは一般にまず、送信側ＳＩＰクライアントのアウトバウンド（ｏｕｔｂｏｕｎｄ）ＳＩＰプロキシに送られる。次いでこのアウトバウンドＳＩＰプロキシがこのリクエストメッセージを、しばしば他の中間ＳＩＰプロキシを介して、意図された受信側クライアントが登録されているＳＩＰプロキシに転送し、次いでこのＳＩＰプロキシが、ＩＮＶＩＴＥメッセージを受信側クライアントに送る。受信側クライアントの受入れメッセージ（「２００ＯＫ」）が、シグナリングチェーンを介して呼出し元クライアントに戻され、これによって、呼出し元クライアントは呼出し先クライアントと、一般にシグナリングチャネルとは別の媒体チャネルを通して通信できるようになる。他のＳＩＰクライアントと通信する他に、ＳＩＰクライアントはさらに、ＲＥＧＩＳＴＥＲリクエストを送ることによってＳＩＰレジストラ（ｒｅｇｉｓｔｒａｒ）に自体を登録するなどの目的で、ＳＩＰサーバと通信することができる。

【０００４】

【発明が解決しようとする課題】さまざまな応用に対して幅広く実装されてはいるが、ＳＩＰは、主としてシグナリングオペレーション用に設計されたものである。ＳＩＰは、通信セッションのセキュリティおよびプライバシーを保護するためのセキュリティ機構を明示的に提供せず、またはこれを必要としない。しかし多くのケースで、ＳＩＰクライアントが、ＳＩＰクライアントのユーザを認証するよう求めるリクエストをアウトバウンドＳＩＰプロキシに送ることを要求し、アウトバウンドＳＩＰプロキシが、ＳＩＰクライアントに対して自体を認証することを要求することが望ましい。さらに、ＳＩＰリクエストメッセージの完全性を保護することもしばしば必要である。クライアント・プロキシ認証およびメッセージの完全性はともに、信頼性の高いセキュリティ機構の使用を要求する。したがって、信頼性の高いセキュリティ機構をＳＩＰシグナリングオペレーションと組み合わせて、ＳＩＰクライアントとアウトバウンドプロキシとの間の認証を可能にすることが求められている。しかし、所望のセキュリティ機構をＳＩＰシグナリングフレームワークにどのようにはめ込んで、目的の異なるこれらの２つの機構を一緒に効果的に実行することができるようにするかということが技術上の課題である。

【０００５】

【課題を解決するための手段】以上のことを考慮して、本発明は、ケルベロス（Ｋｅｒｂｅｒｏｓ）プロトコル、ＮＴＬＭプロトコルなどのセキュリティ機構を、ＳＩＰシグナリングオペレーションのメッセージフローに組み込んで、ＳＩＰクライアントとＳＩＰプロキシが相互に認証できるようにするための方式を提供する。本発明によれば、ＳＩＰクライアントからＳＩＰリクエストメッセージを受け取ると、プロキシはこれに応答して、予め選択されたセキュリティ機構に基づく認証が必要であることを指示するチャレンジメッセージを送る。これに応答してＳＩＰクライアントは、前記セキュリティ機構に基づいてサーバに対してクライアントを認証するための認証データを含むｐｒｏｘｙ　ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有する前記リクエストメッセージの第２バージョンないし改訂バージョンを送る。ケルベロスセキュリティ機構を使用する場合には、ｐｒｏｘｙ　ａｕｔｈｏｒｉｚａｔｉｏｎヘッダが、プロキシにアクセスするためにクライアントが取得したケルベロスサーバチケットを表すデータを含む。ｐｒｏｘｙ　ａｕｔｈｏｒｉｚａｔｉｏｎヘッダのデータに基づくクライアントのユーザの認証に成功した場合、ＳＩＰプロキシは、ＳＩＰクライアントとリクエストメッセージの意図された受信者との間のＳＩＰメッセージシグナリング経路に沿ってこのリクエストを転送する。ＳＩＰクライアントが相互認証を要求している場合、ＳＩＰプロキシは、クライアントに送る次のメッセージにｐｒｏｘｙ　ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎ（プロキシ認証情報）ヘッダを追加する。このメッセージは例えば、ＩＮＶＩＴＥリクエストに応答して呼出し先ＳＩＰクライアントが生成した「２００ＯＫ」ＳＩＰ応答、またはＲＥＧＩＳＴＥＲメッセージに応答してＳＩＰレジストラサーバが生成した「２００ＯＫ」応答である。ｐｒｏｘｙａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎヘッダは、クライアントがＳＩＰプロキシを認証するための認証データを含む。

【０００６】本発明の特徴は請求項に詳細に記述されているが、本発明、ならびに本発明の目的および利点は、添付図面を参照して以下の詳細な説明を読むことによって最もよく理解されよう。

【０００７】

【発明の実施の形態】図面を参照すると、適当なコンピューティング環境中に実現された本発明が示されている。図面中、同様の参照符号は同様の要素を指す。そうでなければならないというわけではないが、本発明は、パーソナルコンピュータによって実行される、プログラムモジュールなどのコンピュータ実行可能命令の一般的な文脈で説明される。プログラムモジュールには一般に、特定のタスクを実行し、または特定の抽象データ型

を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。さらに、本発明を、ハンドヘルド装置、マルチプロセッサシステム、マイクロプロセッサベースの、またはプログラム可能な家庭用電子機器、ネットワークＰＣ、ミニコンピュータ、メインフレームコンピュータなどを含む他のコンピュータシステム構成とともに実施できることを当業者は理解されたい。本発明は、通信ネットワークを介してリンクされた遠隔処理装置によってタスクが実行される分散コンピューティング環境で実施することができる。分散コンピューティング環境では、プログラムモジュールが、ローカルメモリ記憶装置と遠隔メモリ記憶装置の両方に位置することができる。

【０００８】以下の説明ではまず、本発明を実現するための例示的なシステム中で使用することができる汎用コンピューティング装置を説明し、次いで、図２～９を参照して本発明をより詳細に説明する。図１を参照すると、処理ユニット２１、システムメモリ２２およびシステムバス２３を含む従来のパーソナルコンピュータ２０の形態の汎用コンピューティング装置が示されている。システムバス２３は、システムメモリを含むさまざまなシステムコンポーネントを処理ユニット２１に結合する。システムバス２３は、さまざまなバスアーキテクチャのうちの任意のアーキテクチャを使用した、メモリバスまたはメモリコントローラ、周辺バスおよびローカルバスを含むいくつかのタイプのバス構造のうちの任意の構造とすることができる。システムメモリは、リードオンリーメモリ（ＲＯＭ）２４およびランダムアクセスメモリ（ＲＡＭ）２５を含む。ＲＯＭ２４には、起動時などにパーソナルコンピュータ２０の内部要素間の情報転送を助ける基本ルーチンを含む基本入出力システム（ＢＩＯＳ）２６が記憶されている。パーソナルコンピュータ２０はさらに、ハードディスク６０の読取り／書込み用のハードディスクドライブ２７、リムーバブル磁気ディスク２９の読取り／書込み用の磁気ディスクドライブ２８、およびＣＤ－ＲＯＭまたは他の光媒体などのリムーバブル光ディスク３１の読取り／書込み用の光ディスクドライブ３０を含む。

【０００９】ハードディスクドライブ２７、磁気ディスクドライブ２８および光ディスクドライブ３０はそれぞれ、ハードディスクドライブインタフェース３２、磁気ディスクドライブインタフェース３３、および光ディスクドライブインタフェース３４によってシステムバス２３に接続されている。これらのドライブおよび関連コンピュータ可読媒体は、コンピュータ可読命令、データ構造、プログラムモジュール、およびパーソナルコンピュータ２０のためのその他のデータを記憶する不揮発性記憶装置を提供する。ここに記載した例示的な環境では、ハードディスク６０、リムーバブル磁気ディスク２９およびリムーバブル光ディスク３１が使用されているが、この例示的な動作環境では、磁気カセット、フラッシュメモリカード、ディジタルビデオディスク、ベルヌーイカートリッジ、ランダムアクセスメモリ、リードオンリーメモリ、ストレージエリアネットワークなど、コンピュータがアクセス可能なデータを記憶することができる他のタイプのコンピュータ可読媒体を使用することもできることを当業者は理解されたい。

【００１０】ハードディスク６０、磁気ディスク２９、光ディスク３１、ＲＯＭ２４またはＲＡＭ２５には、オペレーティングシステム３５、１つまたは複数のアプリケーションプログラム３６、他のプログラムモジュール３７およびプログラムデータ３８を含む、いくつかのプログラムモジュールを記憶することができる。ユーザは、キーボード４０、ポインティングデバイス４２などの入力装置を介して、コマンドおよび情報をパーソナルコンピュータ２０に入力することができる。この他の入力装置（図示せず）には例えば、マイクロホン、ジョイスティック、ゲームパッド、衛星アンテナ、スキャナなどがある。これらの入力装置および他の入力装置はたいてい、システムバスに結合されたシリアルポートインタフェース４６を介して処理ユニット２１に接続されるが、パラレルポート、ゲームポート、ユニバーサルシリアルバス（ＵＳＢ）、ネットワークインタフェースカードなどの他のインタフェースによって接続することもできる。さらに、モニタ４７または他のタイプのディスプレイ装置が、ビデオアダプタ４８などのインタフェースを介してシステムバス２３に接続されている。モニタの他に、パーソナルコンピュータは一般に、スピーカ、プリンタなどの周辺出力装置（図示せず）を含む。

【００１１】パーソナルコンピュータ２０は、リモートコンピュータ４９などの１台または数台のリモートコンピュータへの論理接続を使用したネットワーク化された環境で動作することができる。リモートコンピュータ４９は、別のパーソナルコンピュータ、サーバ、ルータ、ネットワークＰＣ、ピア装置または他の普通のネットワークノードとすることができる。図１にはメモリ記憶装置５０だけしか示されていないが、リモートコンピュータ４９は一般に、パーソナルコンピュータ２０に関して先に説明した全ての要素または多くの要素を含む。図１に示した論理接続は、ローカルエリアネットワーク（ＬＡＮ）５１およびワイドエリアネットワーク（ＷＡＮ）５２を含む。このようなネットワーキング環境は、事業所、企業内コンピュータネットワーク、イントラネットおよびインターネットでごく普通に見られるものである。

【００１２】ＬＡＮネットワーキング環境で使用するとき、パーソナルコンピュータ２０は、ネットワークインタフェースまたはアダプタ５３を介してローカルネットワーク５１に接続される。ＷＡＮネットワーキング環境で使用するとき、パーソナルコンピュータ２０は一般に

モデム５４、またはＷＡＮ５２を介した通信を確立するためのその他の手段を含む。モデム５４は、内部モデムでもまたは外部モデムでもよく、シリアルポートインタフェース４６を介してシステムバス２３に接続される。ネットワーク化された環境では、パーソナルコンピュータ２０に関して図示したプログラムモジュール、またはその一部分を、遠隔メモリ記憶装置に記憶することができる。示されたネットワーク接続は例示的なものであり、コンピュータ間に通信リンクを確立する他の手段を使用することができることを理解されたい。

【００１３】特に指示しない限り以下の説明では、１台または数台のコンピュータによって実行されるアクト（ａｃｔ）およびオペレーションの記号的表現を参照して、本発明を説明する。時にコンピュータ実行アクトまたはオペレーションと呼ばれるこのようなアクトおよびオペレーションは、コンピュータの処理ユニットによる、構造化された形態のデータを表す電気信号の操作を含むことを理解されたい。この操作では、データを変換し、またはコンピュータのメモリシステム中の位置にデータを維持する。これによって、コンピュータのオペレーションは当業者によってよく理解されている方法で再構成され、または別な方法で変更される。データが維持されるデータ構造は、データのフォーマットによって定義された特定の特性を有するメモリの物理的位置である。しかし、本発明は以上の文脈で説明されるが、これは限定を意味しない。当業者なら分かるとおり、以降に説明するさまざまなアクトおよびオペレーションはハードウェアとして実装することもできるからである。

【００１４】次に図２を参照する。本発明は、セキュリティ機構、特にケルベロス認証プロトコルを実装したセキュリティ機構を、セッションイニシエーションプロトコル（ＳＩＰ）に基づくリクエストメッセージに組み込んで、ＳＩＰクライアント７２とＳＩＰプロキシサーバ７４が相互に認証することができるようにし、かつシグナリングメッセージの完全性を保護するための方法を対象とする。ＳＩＰは、ＲＦＣ（Ｒｅｑｕｅｓｔ ｆｏｒ Ｃｏｍｍｅｎｔｓ）２５４３に定義されている。この文書は、その全体が参照によって本明細書に組み込まれる。

【００１５】例えば、一般的なセッション開始オペレーションでは図２に示すように、別のユーザ８０（例えば「ボブ（Ｂｏｂ）」）と話しをしたいＳＩＰクライアント７２（「呼出し元」）のユーザ７６（例えば「アン（Ａｎｎ）」）は、ボブがＩＮＶＩＴＥメッセージ８２の意図された受信者であることを確認するＩＮＶＩＴＥメッセージを送る。このＩＮＶＩＴＥメッセージは、呼出し元ＳＩＰクライアントのドメインに対応するアウトバウンドプロキシサーバ７４に送られる。図２に示すようにＩＮＶＩＴＥメッセージは、このシグナリングオペレーションに関与する複数のＳＩＰプロキシを介して送られ、その後、ボブのコンピュータ８８のＳＩＰクライアント８６（呼出し先）に到達する。好ましい実施形態では、シグナリング経路上のＳＩＰプロキシ間で転送中のＳＩＰシグナリングメッセージのセキュリティが、ＩＰｓｅｃプロトコルの下で、またはＳＳＬ（ＳｅｃｕｒｅｄＳｏｃｋｅｔｓ Ｌａｙｅｒ）プロトコルの下でパイプを通してメッセージを送ることによって保護される。この例ではＳＩＰリクエストがＩＮＶＩＴＥリクエストだが、以下で説明する認証方式は、ＲＥＧＩＳＴＥＲ、ＭＥＳＳＡＧＥ、ＳＵＢＳＣＲＩＢＥ、ＳＥＲＶＩＣＥなどの他のタイプのＳＩＰリクエストに対しても使用することができることを理解されたい。

【００１６】シグナリングオペレーションのセキュリティおよびシグナリングメッセージの完全性を確保するため、アウトバウンドＳＩＰプロキシサーバ７４は、シグナリング経路９０を介してＩＮＶＩＴＥメッセージ８２を転送する前に、呼出し元ＳＩＰクライアント７２のユーザ７６の認証を求めることができる。次に図３を参照する。本発明によればプロキシサーバ７４はＩＮＶＩＴＥメッセージに応答して、ＳＩＰクライアント７２にチャレンジメッセージ９６を送る。チャレンジメッセージ９６は、まず最初にクライアント７２がそのユーザをプロキシ７４に対して認証しなければならないことを指示する、ＳＩＰ仕様に定義されたステータスコード「４０７ Ｐｒｏｘｙ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｒｅｑｕｉｒｅｄ（４０７プロキシ認証が必要）」を含む。ＳＩＰ仕様によれば、チャレンジメッセージ９６（以後「４０７メッセージ」と呼ぶ）は、認証のためにクライアントが使用しなければならないセキュリティ機構を指示するデータを含む「Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅ」ヘッダフィールド９８を含む。Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅヘッダの構文および内容ヘッダについては後に詳細に説明する。好ましい実施形態では、ケルベロスが好ましいセキュリティ機構だが、ＳＩＰフレームワークでは、ＮＴＬＭプロトコルに基づくセキュリティ機構の使用も可能である。以下の説明では、特に指示しない限り、ケルベロスプロトコルに基づくセキュリティ機構を使用するものとする。

【００１７】さらに図３を参照する。ＩＮＶＩＴＥメッセージ８２に応答したプロキシサーバ７４から４０７メッセージ９６を受け取ると、Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅヘッダ９８からＳＩＰクライアント７２は、ケルベロス機構を使用したユーザの認証をプロキシサーバが求めていると判断する。サーバチケットをまだ取得していない場合、クライアント７２は次いで、ＳＩＰプロキシサーバ７４のケルベロス・キー・ディストリビューション（配布）・センター（ＫＤＣ：Ｋｅｙ Ｄｉｓｔｒｉｂｕｔｉｏｎ Ｃｅｎｔｅｒ）１００からサーバチケット１０８を取得する。一実施態様では、ＫＤＣ１００が、プロキシサーバ７４のドメインコントロー

ラ１０２の一部である。サーバチケット１０８を取得した後、クライアント７２は別のＩＮＶＩＴＥメッセージ１１０を送る。ただしこのときには、ＩＮＶＩＴＥメッセージ１１０が、ＳＩＰ仕様に基づくＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎ（プロキシ許可）ヘッダフィールド１１２を含んでいる。Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダフィールド１１２は、プロキシサーバにアクセスするためのサーバチケット１０８を含む。サーバチケット１０８は、使用するセッションキー１１６を含んでいる。Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダフィールドの構文および内容については後に詳細に説明する。任意選択で、Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダにさらに、クライアント７２に対して自体を認証するようプロキシサーバ７４に求める、相互認証のためのリクエストを含めることもできる。

【００１８】ＳＩＰプロキシサーバ７４は、ケルベロスサーバチケットが埋め込まれた再送ＩＮＶＩＴＥメッセージ１１０を受け取ると、サーバチケットを抜き出し、チケットの有効性を、ＫＤＣ１００と共有しているその長期キーでチケットを解読することによって検証する。チケットが有効である場合、ユーザ７６は認証され、ＳＩＰプロキシサーバ７４は、シグナリング経路上の次のプロキシ１２０にＩＮＶＩＴＥメッセージ１１０を転送する。クライアント７２が、ＩＮＶＩＴＥメッセージ１１０のＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダ１１２の中で相互認証を要求してきた場合、プロキシサーバ７４は、ケルベロスサーバチケットに関連付けられたセッションキーを使用して、自体からクライアントへの将来のパケットに署名する。このメッセージは、クライアント７２がプロキシ７４を認証することを可能にするプロキシ７４の証明書（ｃｒｅｄｅｎｔｉａｌ）を含むＰｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｉｎｆｏｒｍａｔｉｏｎヘッダ１２４を含む。

【００１９】最終的に、ＩＮＶＩＴＥメッセージ１１０は呼出し先、すなわちボブのコンピュータ８８のＳＩＰクライアント８６に届く。この呼勧誘を受け入れる場合、呼出し先は「２００　ＯＫ」メッセージ１２６を返し、このメッセージは呼出し元へ送られる。呼接続が確立されると、呼出し元は、このシグナリング段階に関与したＳＩＰプロキシを通すことなく、呼出し先と直接に通信することができる。

【００２０】次に図４を参照する。本発明によれば、ＳＩＰクライアント７２とＳＩＰプロキシサーバ７４の間で認証セキュリティアソシエーション（ＳＡ：ｓｅｃｕｒｉｔｙ　ａｓｓｏｃｉａｔｉｏｎ）を確立するオペレーションは、状態機械１２８として見ることができる。図４に示す実施形態では、好ましいセキュリティ機構がケルベロスであるが、任意選択でＮＴＬＭセキュリティ機構も使用することができ、この状態機械図は、この任意選択の包含を反映したものになっている。

【００２１】図４では、状態が円の中に、状態に関連して実行されるオペレーションが長方形のブロックの中に示されている。図４に示すように、状態機械の１つの状態が、セキュリティＳＡが確立されていない「ＳＥＣＵＲＩＴＹ＿ＳＴＡＴＥ＿ＮＯＮＥ」状態１３２である。クライアント７２が送ったＩＮＶＩＴＥメッセージに応答したプロキシ７４から４０７チャレンジメッセージを受け取り、またはプロキシとプレ認証（ｐｒｅ－ａｕｔｈｅｎｔｉｃａｔｉｏｎ）を実行すると決定すると、クライアント７２は「ＳＥＣＵＲＩＴＹ＿ＳＴＡＴＥ＿ＡＣＱＵＩＲＩＮＧ＿ＳＡ」状態１３６に入り、認証に必要なセキュリティアソシエーションデータを獲得する。認証に必要なセキュリティアソシエーションデータは、選択したセキュリティ機構によって異なる。

【００２２】セキュリティアソシエーションは一般に、クライアントとＳＩＰプロキシが共有の秘密を安全な方法で交換し、この秘密を使用して認証し、クライアントとプロキシによって交換される以降のメッセージの完全性を保護することができる状態と定義される。セキュリティ機構がケルベロスである場合、セキュリティアソシエーションは、プロキシに対するケルベロスサーバチケットおよびセッションキーを含む。ケルベロスの場合、取得されたＳＡは完全である。すなわち、これを使用して、ＳＩＰクライアントのユーザをプロキシが十分に認証することができる。クライアントは次いで、このＳＡ関連情報（例えばサーバの秘密を用いて暗号化されたケルベロスセッションキー）をプロキシに送る（段階１３８）。プロキシが、署名された２０００Ｋメッセージを返した場合（段階１４０）、認証は成功であり、セキュリティアソシエーションは確立されている。すなわちクライアントはＳＡ＿Ｅｓｔａｂｌｉｓｈｅｄ状態１４２にある。しかし、プロキシが代わりに４０７チャレンジメッセージを返した場合（段階１４６）、クライアントは、プロキシが不良な状態にあり、そのためクライアントの正当な証明書を検証することができないでいるとみなす。次いでクライアントは「バックオフ」時間（例えば５分）のあいだ待機し（段階１４８）、その後再びＳＩＰメッセージを送る。

【００２３】ＳＡ＿Ｅｓｔａｂｌｉｓｈｅｄ状態１４２に入った後は、セキュリティアソシエーションが期限切れにならない限り、クライアントは再び認証を実行することなくプロキシに別のメッセージを送ることができる。しかしプロキシが４０７チャレンジメッセージを送った場合（段階１５０）、クライアントは、確立されたセキュリティアソシエーションをプロキシが何かの理由で取り下げたとみなす。その結果、クライアントは、ＳＡ＿Ｄｒｏｐｐｅｄ状態１５６に入り、ＳＥＣＵＲＩＴＹ＿ＳＴＡＴＥ＿ＡＣＱＵＩＲＩＮＧ＿ＳＡ状態１３６へ移って、プロキシとの認証を再び実施するための新し

いＳＡを獲得する。

【００２４】先に述べたとおり、ユーザの認証に対して任意選択でＮＴＬＭ機構を選択することができる。ＮＴＬＭの状態移行はケルベロスの状態移行と概ね同じだが、ＮＴＬＭは、始めに不完全なＳＡだけを獲得し（段階１５８）、不完全なＳＡを第１のメッセージに入れてプロキシに送るという違いがある。具体的には、ＮＴＬＭの場合、ＳＡ関連情報を含むクライアントからの第１のリクエストは、クライアントのセキュリティ関連機能（例えばクライアントがサポートするプロトコルのバージョン、サポートする署名アルゴリズムなど）を伝える。これに応答してプロキシサーバは、そのＮＴＬＭ関連機能および一般に「ｎｏｎｃｅ」と呼ばれるランダムなバイト列を含む自体の認証データを含んだ第２の４０７チャレンジを送る（段階１６０）。これに応答してクライアントは、それ自体の名称およびプロキシによって送られた「ｎｏｎｃｅ」値のハッシュにその証明書を使用して署名する。これは、ＮＴＬＭインプリメンテーションによって内部的に処理される。プロキシサーバは、＊クライアントの認証データを検証し、ドメインコントローラの助けを借りてセッションキーを得る。ＳＩＰプロキシが意図された受信者でない場合、プロキシは、シグナリング経路上の次のホップにＳＩＰリクエストを転送し、送信側ＳＩＰクライアントへの次のメッセージ（例えば受信者からの２０００Ｋメッセージ）に署名する（段階１４０）。

【００２５】ＳＩＰクライアントとＳＩＰプロキシの間の認証目的のメッセージ交換に関与するさまざまなＳＩＰヘッダの構文については後に説明する。

【００２６】（４０７応答）先に述べたとおり、ＩＮＶＩＴＥメッセージを送ったＳＩＰクライアント（またはそのユーザ）の識別を要求したい場合、ＳＩＰプロキシサーバ７４は、Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅヘッダを含む４０７メッセージをクライアントに送る。認証のためにケルベロスセキュリティ機構の使用を必要とする好ましい実施形態のＰｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅヘッダの構文は以下のとおりである。

【００２７】

```
Proxy-Authenticate = "Proxy-Authenticate" scheme kerb-challenge gssapi-d
ata
Scheme          = "kerberos" | "NTLM" | "Negotiate"
kerb-challenge    = 1# (realm | targetname | [opaque] | qop-options | gg
ssapi-
data)
targetname       = "targetname" "=" <"> URI (1*SP URI) <">
URI             = absoluteURI | abs_path
opaque           = "opaque" "=" quoted-string
qop-options       = "qop" "=" <"> 1#qop-value <">
qop-value        = "auth" | "auth-int" | token
gssapi-data       = "gssapi-data" "=" (token | quoted-string)
```

【００２８】ここに記載したＰｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅヘッダの構文は、「ＨＴＴＰ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ：Ｂａｓｉｃ　ａｎｄ　ＤｉｇｅｓｔＡｃｃｅｓｓ　Ａｕｔｈｅｎｔｉｃａｔｏｎ」という表題のＩＥＴＦ　ＲＦＣ２６１７に定義されている「ＷＷＷ－Ａｕｔｈｅｎｔｉｃａｔｅ　Ｒｅｓｐｏｎｓｅ　Ｈｅａｄｅｒ」と同様である。この文書はその全体が参照によって本明細書に組み込まれる。任意選択のパラメータ「ａｌｇｏｒｉｔｈｍ」および「ｓｔａｌｅ」は省かれている。ヘッダの「ｓｃｈｅｍｅ」フィールドは、サーバに対して自体を認証するのに、サーバによって提案された認証機構のうちどの認証機構を使用したいかをクライアントが選択することを可能にする。クライアントがケルベロス機構をサポートしている場合には、クライアントはケルベロス機構を選択することが好ましく、サポートしていない場合にはＮＴＬＭ認証機構を選択する。

【００２９】ｒｅａｌｍパラメータは、ＳＩＰプロキシが属し、クライアントがアクセスしようとしているＳＩＰサービスプロバイダの固有の識別子である。認証のためにユーザが提供する必要がある正しい証明書セットをユーザが識別するのを助けるため、ｒｅａｌｍ列はユーザに表示される。「ｔａｒｇｅｔｎａｍｅ」パラメータは常に必要なパラメータであり、ＳＩＰプロキシのＦＱＤＮを伝達するのに使用される。このパラメータの実際の内容は、クライアントがＳＡを確立しているプロキシを追跡するのを助ける。それは、応答が自体に対するものなのか、または他のプロキシに対するものなのかをプロキシが判定するのを助ける。「ｏｐａｑｕｅ」パラメータは、サーバが、確立されている特定のＳＡに索引をつけるのに使用され、後に説明するように、クライアントがＳＡに対して生成する将来のＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダ中に反映されなければならない。

【００３０】この実施形態では、ＩＥＴＦ　ＲＦＣ２０７８（その全体が参照によって本明細書に組み込まれる）に定義されたＧＳＳ－ＡＰＩ（Ｇｅｎｅｒｉｃ　Ｓｅｃｕｒｉｔｙ　Ｓｅｒｖｉｃｅ　Ａｐｐｌｉｃａｔｉ

ｏｎ ＰｒｏｇｒａｍｍｉｎｇＩｎｔｅｒｆａｃｅ）が実装されており、通信しているアプリケーション間でメッセージを安全に交換するのに使用されるとみなす。ＧＳＳ－ＡＰＩは特に、通信アプリケーションが、別のアプリケーションに関連付けられたユーザを認証することを可能にする。Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅヘッダおよび後述するＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダのｇｓｓａｐｉ－ｄａｔａフィールドは、ＮＴＬＭおよびケルベロスセキュリティパッケージを実装したセキュリティＡＰＩによって、ＳＡネゴシエーション段階の間に返されたデータを保持するためのものである。これらのＡＰＩは、クライアントからプロキシへ、およびプロキシからクライアントへ送る必要があるｇｓｓａｐｉデータを返す。ｇｓｓａｐｉデータは、ＳＩＰクライアント／プロキシインプリメンテーションにとって不透明であり、セキュリティＡＰＩだけが解釈することができる。ｑｏｐパラメータは、クライアントが従って欲しいとサーバが欲しているセキュリティのレベルをクライアントに教える。ｑｏｐパラメータ値は常に、この機構によって提供されるセキュリティレベルがユーザの認証であることを指示する「ａｕｔｈ」にセットされている。

【００３１】Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅヘッダフィールドの一例を以下に示す。

【００３２】

```
Proxy-Authenticate: Negotiate
realm = "Microsoft RTC Service provider",
opaque = "ABCDEF456789"
*qop = "auth",
gssapi-data = "ABCD345678yuikjhlbcdfsaqwety"
```

【００３３】許可されたクライアントだけを許すとされ、クライアントから着信するＳＩＰパケットが署名を含まない場合には一般に、ＳＩＰプロキシは、ＳＩＰクライアントの識別を要求するであろう。（リブートなどのため）ＳＩＰプロキシがこのＳＩＰ ＵＲＩに対するセキュリティアソシエーションを失った場合にも、ＳＩＰプロキシはクライアントの認証を要求するであろう。クライアントが使用している許可パラメータとＳＩＰプロキシが期待しているものの間にミスマッチがある場合、ＳＩＰプロキシは、クライアントが従って欲しいとＳＩＰプロキシが欲している正確な許可パラメータを運んでいる４０７メッセージを使用して、クライアントの認証を要求するであろう。

【００３４】（４０７チャレンジに対するクライアントの応答）４０７チャレンジに応答してＳＩＰクライアントは、４０７チャレンジメッセージを介してＳＩＰプロキシから送られた認証パラメータに従った署名を生成しようとする。ＳＩＰクライアントはＣｓｅｑ値を増分し、チャレンジの対象となった最初のＳＩＰリクエストを、Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎリクエストヘッダの中に入れられた許可情報とともに再送する。好ましい実施形態におけるＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎリクエストヘッダの構文は以下のとおりである。

【００３５】

```
Proxy-Authorization = "Proxy-Authorization" ":" scheme kerbresponse real
m
message-qop targetname
     kerb-response = 1# ( [crand) | [response] | [opaque] | [gssapi-data
] )
        message-qop = "qop" "=" qop-value
        crand       = "crand" "=" crand-value
        crand-value  = crand-value
        response    = "response" "=" request-digest
        request-digest = <"> 32LHEX <">
        LHEX      = "0" | "1" | "2" | "3" |
                    "4" | "5" | "6" | "7" |
                    "8" | "9" | "a" | "b" |
                    "c" | "d" | "e" | "f" |
```

【００３６】ここに記載したＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダの構文は、ＩＥＴＦ ＲＦＣ ２６１７に定義されている「Ａｕｔｈｏｒｉｚａｔｉｏｎ Ｒｅｑｕｅｓｔ Ｈｅａｄｅｒ」と同様である。ただし、任意選択のパラメータ「ａｌｇｏｒｉｔｈｍ」および「ＵＲＩ」は省かれている。Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダは、リクエストＵＲＩおよびＶｉａヘッダの後に追加される。署名は、セッションキーを使用して、以下のフィールドにわたって計算される。

【００３７】

－ＦｒｏｍヘッダＵＲＩ
－ＴｏヘッダＵＲＩ
－Ｆｒｏｍヘッダタグ
－Ｔｏヘッダタグ
－Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダ中の「ｃｒａｎｄ」パラメータ、またはＰｒｏｘｙ－Ａｕｔ

ｈｅｎｔｉｃａｔｉｏｎ－Ｉｎｆｏヘッダ中の「ｓｒａｎｄ」パラメータ
－ＳＩＰメッセージＥｘｐｉｒｅｓヘッダ中のＥｘｐｉｒｅｓ値
署名には、ＳＩＰメッセージのメッセージ本体は含まれていない。ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダは、ｇａｓａｐｉ－ｄａｔａパラメータまたはｒｅｓｐｏｎｓｅ（署名）パラメータを含む。

【００３８】以下に、４０７チャレンジに対するクライアントの応答におけるＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダの例を示す。

【００３９】

```
Proxy-Authorization: Negotiate
realm = "Microsoft RTC Service Provider",
response = "ABCD87654cvx,
opaque = "ABCD1234",
crand = "1234"
qop = "auth"
targetname = "server1.domainA.mjcrosoft.com"
```

または

```
Proxy-Authorization: Negotiate
realm = "Microsoft RTC Service Provider",
opaque= "ABCD1234",
gssapi-data = "ABCDEF123456",
qop = "auth",
*targetname = "server1.domainA.microsoft.com"
```


【００４０】プロキシからの４０７チャレンジに応答するときの他に、クライアントは、自体を初めてＳＩＰプロキシに登録するときにもこのヘッダを送る。ＳＩＰクライアントが自体をプロキシサーバに登録し、セッションのためにセキュリティアソシエーションを初期化するプロセスにあるとき、Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダは、「ｇｓｓａｐｉ－ｄａｔａ」パラメータを含む。

【００４１】（相互認証）ある種のシナリオでは、ＳＩＰプロキシとＳＩＰクライアントの間で相互認証を確立することが必要である。クライアントは、特定のプロキシサーバに対してクライアントが有する準備プロファイルから、相互認証が必要か否かを判断する。相互認証が使用可能な場合、クライアントは、ＧＳＳ　ＡＰＩの標準バージョンを使用して相互認証のためのセキュリティアソシエーションを初期化する。相互認証が使用可能な場合にはさらに、サーバが、ＳＩＰクライアントに送る全てのパケットに署名する必要がある。この署名は、Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｉｏｎ－Ｉｎｆｏｒｍａｔｉｏｎリクエストヘッダに入れて運ばれる。Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅ－Ｉｎｆｏｒｍａｔｉｏｎの構文は以下のとおりである。

【００４２】

```
ProxyAuthenticationInfo = "Proxy-Authentication-Info" ":" auth-info
    auth-info    = 1# (message-qop | response-auth | srand)
       response-auth = "rspauth" "=" response-digest
    response-digest = <"> *LHEX <">
    srand        = "srand" "=" srand-value
    srand-value = quoted-string
```

【００４３】Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｉｏｎ－Ｉｎｆｏヘッダ中の「ｒｓｐａｕｔｈ」パラメータは、この応答に対する（認証を求めているプロキシの）署名を運ぶ。「ｓｒａｎｄ」パラメータは、ＳＡ確立段階後にサーバが、クライアントに送るメッセージに署名するのに使用される。このパラメータは、サーバによって生成されるランダムな文字列であり、生成されたメッセージのハッシュ／署名にランダム性の要素を導入するのに使用される。

【００４４】以下に、Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｉｏｎ－Ｉｎｆｏｒｍａｔｉｏｎヘッダの一例を示す。

【００４５】

```
Proxy-Authentication-Info: Negotiate
realm = "Microsoft RTC Service Provider",
qop  = "auth",
rspauth = "ABCD87564cvx",
srand = "9876543210",
targetuame = "server1.domainA.microsoft.com"
```

【００４６】ＳＩＰフレームワークでは一般に、ＲＥＧＩＳＴＥＲリクエストを使用した登録プロセスの間に、ＳＩＰクライアントが、ＳＩＰプロキシとのセキュリティアソシエーションを確立することができる。登録によってＳＩＰクライアントは、ＳＩＰプロキシからメッセージを受け取ることができるようになる。ＳＩＰクライアントが自体をＳＩＰプロキシに登録するときには、ＳＩＰクライアントは同時に、ケルベロスチケットなどの認証データをＲＥＧＩＳＴＥＲメッセージに入れて送ることによって、自体のユーザをＳＩＰプロキシサーバに対して認証することができる。ＳＩＰクライアントがＳＩＰプロキシにすでに登録されており、かつＳＩＰプロキシに対して認証されている場合、ＩＮＶＩＴＥなどのＳＩＰリクエストをクライアントが送るときには、クライアントからのリクエストメッセージが、ＳＡ確立プロセスの間に交換されたケルベロスセッションキーを使用して署名される。

【００４７】とは言え、自体をサーバに登録しなくても、ＳＩＰクライアントはリクエストメッセージをＳＩ

Ｐプロキシに送ることができる。呼出し元がプロキシに対して認証されていない場合には（たとえＳＩＰクライアントがプロキシに登録されている場合であっても）、ＳＩＰプロキシはそのリクエストを次のホップに転送しない。その代わりにプロキシは、ＳＩＰクライアントにチャレンジを送る。

【００４８】このチャレンジは、クライアントがこのＳＩＰサーバと、セキュリティアソシエーションを確立する必要があることを指示する。クライアントは、セキュリティアソシエーションデータを含むリクエストを再び送ることによってＳＡを確立することができ、あるいは、登録はすでにされているが、ＳＡがまだ確立されていない場合には、このサーバへの自体の登録をリフレッシュすることによってＳＡを確立することができる。登録リフレッシュを使用してＳＡを確立し、次いで有効な署名を有するＳＩＰリクエストを送ることには、登録が良好な状態にあることが保証されるという利点がある。

【００４９】さらに、ＳＩＰクライアントがＳＩＰプロキシに対する登録を抹消するたびに、ＳＩＰクライアントとＳＩＰプロキシの間のセキュリティアソシエーション（ＳＡ）は失われ、新しいセキュリティアソシエーションを再び交渉し直さなければならない。さらに、ＳＩＰクライアントの登録が失効すると、プロキシサーバは、対応するセキュリティコンテキストをＳＡのリストから削除する。登録をリフレッシュするたびに、ＳＩＰクライアントは、認証セキュリティアソシエーションもリフレッシュしなければならない。

【００５０】ケルベロスプロトコルに基づくセキュリティ機構を使用する好ましい実施形態では、送信側ＳＩＰクライアントのユーザの認証がＳＩＰプロキシ／レジストラによって要求された場合に、ＳＩＰクライアントがＳＩＰプロキシに登録するたびに、ケルベロスキーディストリビューションセンター（ＫＤＣ）からのケルベロスチケットが要求される。ケルベロスチケットを受け取ると、ＳＩＰクライアントはこのチケットを解読する。解読されたチケットは、セッションキーおよびこのケルベロスセッションの他のいくつかの特性を含む。このチケットはさらに、サーバの証明書で暗号化されたセッションキーおよび他のセッション関連パラメータを含む。この部分は、ｇｓｓａｐｉ－ｄａｔａフィールドの、ｐＯｕｔｐｕｔパラメータに入れて返され、ｒｅ－ＩＮＶＩＴＥリクエストに入れてプロキシに送られる。

【００５１】ＳＩＰフレームワーク内でのセキュリティ機構のオペレーションの明瞭な理解を容易にするため、クライアント・プロキシのケルベロス認証の具体的な例を図２を参照して説明する。この例では、ＳＩＰプロキシサーバ７４が、ドメイン「ｄｏｍａｉｎＡ．Ｍｉｃｒｏｓｏｆｔ．ｃｏｍ：Ｓ＿ｓｅｒｖｅｒ１」のＫＤＣ１７０との共有秘密鍵をすでに生成していると仮定する。この例では「ｓｅｒｖｅｒ１」が、ＳＩＰプロキシ／レジストラのコード名として使用される。ＫＤＣ１７０は、プロキシサーバ７４がｓｅｒｖｅｒ＿ＩＤ＝ｓｅｒｖｅｒ１．ｄｏｍａｉｎＢ．ｍｉｃｒｏｓｏｆｔ．ｃｏｍであると知っている。プロキシサーバ７４はさらに、証明書ハンドルを獲得して、クライアントから来る認証リクエストに応答する準備を整える。サーバ証明書は、サーバ認証または相互認証をサポートしているセキュリティプロトコルにおいてＳＩＰクライアント７４に対してプロキシサーバ７４を認証するのに使用される。プロキシサーバ７４は、このサーバを起動するのに使用されるサービスアカウントによって定義されるその証明書に対するハンドルを取得する。サーバはこのハンドルを、ＳＳＰＩ（Ｓｅｃｕｒｉｔｙ　Ｓｕｐｐｏｒｔ　Ｐｒｏｖｉｄｅｒ　Ｉｎｔｅｒｆａｃｅ）の関数ＡｃｑｕｉｒｅＣｒｅｄｅｎｔｉａｌｓＨａｎｄｌｅを呼び出すことによって取得する。

【００５２】図２の例では、ＳＩＰクライアント７２のユーザ７６がアンである。アンは、ＮＴドメインにアカウントを有し、一日の始まりに、以下の情報を使用して自分のアカウントにログオンする。

【００５３】

```
UserID / principal name = ann@microsoft.com
Preferred_email  = ann@microsoft.com
User_domain = domainA.Microsoft.com
Workstation = ann1.domainA.Microsoft.com
```

【００５４】ボブと通話したいとき、アンは、自分のワークステーション７８上のＳＩＰクライアント７２を起動する（ＳＩＰクライアントはサービスとして自動的に起動することができる。ただしユーザのセキュリティコンテキストで走らなければならない）。ＳＩＰクライアント７２は、ＤＮＳを使用してアウトバウンドプロキシサーバ７４を見つける。この例で使用するアウトバウンドプロキシサーバ７４は、Ｓｅｒｖｅｒ１．ｄｏｍａｉｎＢ．Ｍｉｃｒｏｓｏｆｔ．ｃｏｍとして識別される。アンは、自分がｂｏｂ＠ｍｉｃｒｏｓｏｆｔ．ｃｏｍと通話したいことを指示する。アンのＳＩＰクライアント７２は次いで、Ｓｅｒｖｅｒ１．ｄｏｍａｉｎＢ．Ｍｉｃｒｏｓｏｆｔ．ｃｏｍにＩＮＶＩＴＥメッセージ８２を送る。ＩＮＶＩＴＥメッセージは以下の情報を含む。

【００５５】

```
INVITE bob@microsoft.com
From: ann@microsoft.com
To: bob@microsoft.com
```

【００５６】この例の説明を簡潔かつ明瞭に保つため、このＩＮＶＩＴＥメッセージまたはシグナリング処理で交換される他のメッセージに含まれる全てのデータを示すわけではない。ＳＩＰプロキシサーバ７４は、Ｍｉｃｒｏｓｏｆｔ．ｃｏｍのユーザ名空間に対してなされた呼に対する全てのＩＮＶＩＴＥリクエストが認証されることを必要とするように構成されている。その結果、Ｓ

ＩＰプロキシサーバ７４はＩＮＶＩＴＥリクエストに応答して、ケルベロスを使用してユーザ（アン）を認証するようＳＩＰクライアント７４に求める４０７メッセー＊ジ９６を送る。４０７メッセージは以下のデータを含む。

【００５７】

```
Proxy-Authenticate: Kerberos realm = domainB.microsoft.com
targetname = "server1.domainA.Microsoft.com" opaque = "someopaquedata"
```

【００５８】ｏｐａｑｕｅ値は、この呼に対して使用するセキュリティコンテキストを識別するためにプロキシによって初期化される。そのために、プロキシサーバ７４はこのとき、関数ＡｃｃｅｐｔＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔを呼び出し、ｐＯｕｔｐｕｔをｂａｓｅ６４でコード化した結果をｏｐａｑｕｅに入れて返す。クライアントおよびサーバは、このｏｐａｑｕｅ値を使用して、認証の継続、またはＡｕｔｈｏｒｉｚａｔｉｏｎリクエストヘッダを使用した同じサーバへの以降のリクエストの再認証のため特定のサーバに対して使用するセキュリティコンテキストを識別する。

【００５９】アンのワークステーション上のＳＩＰクライアント７２は、認証が必要であることを指示する４０７メッセージ９６を受け取ると、Ｓｅｒｖｅｒ１．ｄｏｍａｉｎＢ．Ｍｉｃｒｏｓｏｆｔ．ｃｏｍと交信するための有効なセッションキーを持っているかどうかをチェックする。まだ所有していない場合には、このドメイン内のＫＤＣと接触して、アウトバウンドＳＩＰプロキシにアクセスするのためのセッションキーを取得する必要がある。この例では、４０７メッセージの中に指定されたｒｅａｌｍからクライアントは、自体のドメインとは異なるドメインにプロキシがあることを知る。

【００６０】プロキシサーバ７４への安全な接続を確立するため、クライアント７２は、認証リクエストをプロキシに送る前にアウトバウンド証明書ハンドルを獲得する。これは、関数ＳＳＰＩを呼び出すことによって実行される。ＳＳＰＩは、ネットワーク化されたアプリケーションが、いくつかあるうちの１つのセキュリティサポートプロバイダ（ＳＳＰ）にコールして、認証済みの接続を確立し、確立された接続を介してデータを安全に交換する手段を提供する。認証セットアップに関与する２つのクライアント側ＳＳＰＩ関数がある。ＡｃｑｕｉｒｅＣｒｅｄｅｎｔｉａｌｓＨａｎｄｌｅ関数は、以前に取得したログオン証明書への参照を取得する。関数ＩｎｉｔｉａｌｉｚｅＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔは、最初の認証リクエストセキュリティトークンを生成する。ＩｎｉｔｉａｌｉｚｅＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔを呼び出すと、４０７メッセージから取得したｏｐａｑｕｅ値がｐＩｎｐｕｔに入れて渡される。クライアントは、この関数のｔｆＣｏｎｔｅｘｔＲｅｑパラメータをリクエストＭＵＴＵＡＬ＿ＡＵＴＨにセットする。ｐｆＣｏｎｔｅｘｔＡｔｔｒポインタは、ｍｕｔｕａｌ－ａｕｔｈが「リクエスト」されたことをケルベロスモジュール１８０がクライアントに知らせる方法である。この情報は、クライアントのケルベロスモジュール１８０によって生み出されるＫＥＲＢ＿ＡＳ＿ＲＥＱの一部であり、クライアントが相互認証を求めていることをサーバ（ここではＳＩＰプロキシ）に知らせるｓｅｃＢｕｆｆｅｒ（ｐＯｕｔｐｕｔ）に入れて渡される。これはＫＥＲＢリクエストの一部なので、ＳＩＰ機構（ヘッダ／パラメータ）が相互認証を要求する必要はない。

【００６１】図２に示した例では、ＡＰＩ関数ＩｎｉｔｉａｌｉｚｅＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔを呼び出すことによって、以下のケルベロス論理が生じる。最初にクライアント７２が、ＤｏｍａｉｎＢ（ドメインＢ）のプロキシサーバ７４へのサーバチケットをクライアント７２に与えるよう、ｄｏｍａｉｎＡ．Ｍｉｃｒｏｓｏｆｔ．ｃｏｍドメインのＫＤＣ１７０に求める。ｄｏｍａｉｎＡ．Ｍｉｃｒｏｓｏｆｔ．ｃｏｍのＫＤＣ１７０はクライアント７２に、ｃｏｒｐ．Ｍｉｃｒｏｓｏｆｔ．ｃｏｍのＫＤＣ１７２への照会チケットを送る。この照会チケットは、この２つのＫＤＣによって共有されたドメイン間キーの中に暗号化されている。この照会チケットを使用して、クライアントは、ＤｏｍａｉｎＢにあるサーバへのサーバチケットをクライアントに与えるよう、ｃｏｒｐ．Ｍｉｃｒｏｓｏｆｔ．ｃｏｍのＫＤＣ１７２に求める。

【００６２】これに応答してＫＤＣ１７２は、ＤｏｍａｉｎＢのＫＤＣ１７６への照会チケットをクライアントに送る。このチケットは、ＫＤＣ１７２がＤｏｍａｉｎＢのＫＤＣ１７６と共有するドメイン間キーの中に暗号化されている。クライアントは次いで、ＤｏｍａｉｎＢにあるプロキシサーバ７４へのチケットをクライアントに与えるよう、ＤｏｍａｉｎＢのＫＤＣ１７６に求める。ＫＤＣ１７６は、プロキシサーバ７４にアクセスするためのサーバチケット１０８をクライアントに送る。ＫＤＣ１７６は、アンのログオンセッションキーを用いてこのセッションキーの１つのコピーを暗号化し、セッションキーの別のコピーをアンの許可データとともにサーバチケットに埋め込み、プロキシサーバの長期キーを用いてサーバチケットを暗号化する。ＫＤＣ１７６は次いで、これらの証明書を、Ｋｅｒｂｅｒｏｓ Ｔｉｃｋｅｔ－Ｇｒａｎｔｉｎｇ Ｓｅｒｖｉｃｅ Ｒｅｐｌｙ（ＫＲＢ＿ＴＧＳ＿ＲＥＰ）に入れてクライアント７２に送る。

【００６３】このように、ＩｎｉｔｉａｌｉｚｅＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔを呼び出すことによって、クライアントマシンのケルベロスモジュール１８０がＫＤＣとのＴＧＳ交換を開始する。この交換によって返される値は、プロキシに送るメッセージに署名するためのセ

25

ッションキーである。

【００６４】その後、ＳＩＰクライアント７２は、ＳＩＰプロキシに送る新しいＩＮＶＩＴＥメッセージ１１０（「ｒｅ－ＩＮＶＩＴＥ」メッセージとも呼ばれる）を生成する。この新しいＩＮＶＩＴＥメッセージ１１０は、クライアントがＫＤＣ１７６から受け取ったサーバチケットを含むＧＳＳ－ＡＰＩデータをその中に含ん

26

＊だ、先に説明したｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを含む。セッションキーは、ＩｎｉｔｉａｌｉｚｅＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔ呼出しによって返されたｐＯｕｔｐｕｔバッファの中に入れて返された値である。したがって、新しいＩＮＶＩＴＥメッセージ１１０は以下のデータを含む。

【００６５】

```
INVITE bob@microsoft.com
From: ann@microsoft com
To: bob@microsoft.com
Proxy-authorization: gss-scheme opaque gssapi -rdata
Opaque = someopaquedata
Gssapi-rdata = base64{pOutput} = session key to the proxy
```

【００６６】このＩＮＶＩＴＥメッセージは、プロキシサーバに対してＫＲＢ＿ＡＰ＿ＲＥＱと等価の内容を実行する。

【００６７】メッセージの完全性を保護し、自体を認証する（すなわちメッセージの出処を証明する）ため、クライアントは、セッションキーを用いてＩＮＶＩＴＥメッセージ１１０に署名する。さもないと、第三者がこのＩＮＶＩＴＥメッセージをかぎつけ、ＯｐａｑｕｅおよびＧｓｓａｐｉ－ｄａｔａ値を手に入れ、偽のＩＮＶＩＴＥメッセージを同じサーバに送って、この第三者とそれが選択した任意の転送先との間で通話をおこなう可能性がある。このことは、サーバへのセッションキーが有効である間（デフォルトでは８時間）、クライアントの認証を「盗む」ことができることを意味する。ＩＮＶＩＴＥメッセージに署名することで、第三者がＯｐａｑｕｅおよびＧｓｓａｐｉ－ｒｄａｔａを取り出すことを防ぐことはできないが、新しいＩＮＶＩＴＥメッセージを生成して好き勝手に通話することは防ぐことができる。この問題を回避するためには、サーバが、署名されたリクエストしか受け入れないように構成されていなければならない。

【００６８】クライアント７２は、ＭａｋｅＳｉｇｎａｔｕｒｅ　ＡＰＩを使用し、これを呼び出して、（４０７メッセージのｏｐａｑｕｅ中に識別された）この呼出しで使用されるセキュリティコンテキストにｐｈＣｏｎｔｅｘｔをセットし、その内容を渡してｐＭｅｓｓａｇｅに署名する。この呼出しの出力は、ｐＭｅｓｓａｇｅの中に返された署名されたメッセージである。クライアントは、この署名をＩＮＶＩＴＥ１１０に追加する。再送されたＩＮＶＩＴＥメッセージ１１０を受け取ると、プロキシサーバ７４は、Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダの中のｏｐａｑｕｅ値をチェックし、所与のｐｈＣｏｎｔｅｘｔ値（所与のセキュリティコンテキストへのハンドル）と相関させる。プロキシサーバ７４は、ｇｓｓａｐｉ－ｒｄａｔａを取り出し、ＡｃｃｅｐｔＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔＡＰＩ関数を呼び出し、ｐｒｏｘｙ－ａｕｔｈｏｒｉｚａｔｉｏｎヘッダから得たｇｓｓａｐｉ－ｒｄａｔａ値をこのＡＰＩ関数のｐＩｎｐｕｔ成分の中に渡すことによって、これを自体のケルベロスモジュール１８２を渡す。ケルベロスモジュール１８２は、プロキシの長期キーを使用してサーバチケットを解読し、アンの許可データおよびセッションキーを抜き出す。ケルベロスモジュール１８２は、セッションキーを使用してアンのオーセンティケータ（ａｕｔｈｅｎｔｉｃａｔｏｒ）を解読し、次いで中のタイムスタンプを評価する。

【００６９】オーセンティケータがこのテストにパスした場合、ケルベロスモジュール１８２は、クライアントのリクエストの中の相互認証フラグを探す。フラグがセットされている場合、ケルベロスモジュール１８２はセッションキーを使用して、アンのオーセンティケータからの時刻を暗号化し、その結果を、ＫｅｒｂｅｒｏｓＡｐｐｌｉｃａｔｉｏｎ　Ｒｅｐｌｙ（ＫＲＢ＿ＡＰ＿ＲＥＰ）の中に返す。これによって、ＡｃｃｅｐｔＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔが呼び出され、ＳＥＣ＿Ｅ＿ＯＫ戻り値が返され、オーセンティケータはｐＯｕｔｐｕｔバッファを使用してこのＡＰＩを通過する。ユーザが認証されると、ＳＩＰプロキシ／レジストラはこのリクエストを処理し、ＩＮＶＩＴＥメッセージを、ＳＩＰシグナリング経路上の次のホップに転送する。

【００７０】プロキシのＳＩＰコンポーネントは次いで、ＳＩＰクライアントに転送する次のメッセージを使用してプロキシのオーセンティケータをクライアントに渡し、これによってクライアントがサーバを認証できるようにする。示された例では、このメッセージが「２００ＯＫ」メッセージである。このメッセージは、ＳＩＰプロキシによって生成されたものではない。この２００応答は、ＩＮＶＩＴＥリクエストに応答して呼出し先が生成する。ＳＩＰプロキシは単に、呼出し元に転送する前にセッションキーを用いてこの応答に署名するだけである。

【００７１】先に説明したとおり、オーセンティケータは、Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｉｏｎ－Ｉｎｆｏｒｍａｔｉｏｎヘッダの中にある。このヘッダはさ

らに、クライアントが、この応答を正しいセキュリティコンテキストと一致させるためのｏｐａｑｕｅ値を含む。

【００７２】アンのワークステーション上のＳＩＰクライアント７２は、「２０００Ｋ」メッセージを受け取ると、Ｐｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｉｏｎ－Ｉｎｆｏｒｍａｔｉｏｎヘッダを抜き出し、ＩｎｉｔｉａｌｉｚｅＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔを呼び出す。ｐｈＣｏｎｔｅｘｔ値はｏｐａｑｕｅの中の値にセットされ、ｐＩｎｐｕｔバッファはｒｅｓｐｏｎｓｅ－ｄｉｇｅｓｔにセットされる。クライアント上のケルベロスモジュール１８０は、プロキシと共有するセッションキーを用いてプロキシのオーセンティケータを解読し、プロキシによって返された時刻を、クライアントのオリジナルのオーセンティケータの中の時刻と比較する。２つの時刻が一致した場合、このＩｎｉｔｉａｌｉｚｅＳｅｃｕｒｉｔｙＣｏｎｔｅｘｔの呼出しはＳＥＣ＿Ｅ＿ＯＫを返し、クライアントは、プロキシが本物であることを知る。一致しない場合にはクライアントはこの呼を取りやめなければならない。クライアントは、要求したことをサーバが実行すると信じることができないので、ＣＡＮＣＥＬを送ってこの呼を強制終了せざるを得ない。

【００７３】上に説明した例では、認証が、まず最初にＳＩＰクライアントが認証データを含まないＩＮＶＩＴＥを送り、次いで、認証が必要であることを指示するプロキシからの４０７メッセージに応答して認証データを別のＩＮＶＩＴＥに入れて送るシナリオで実施される。その代わりにクライアントは、プロキシに送る最初のＩＮＶＩＴＥに必要な認証データを含めることもできる。そのためにクライアント７２は、ＳＩＰの下で通話するためにユーザが使用する前に、プロキシに対するサーバチケットをＫＤＣ１７６から取得する。次いで必要な認証データを、先に説明したＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎリクエストヘッダに入れる。こうすることによって、プロキシがクライアントに４０７チャレンジを送って認証データを要求する必要がなくなる。さらに、先に説明した認証オペレーションの例ではＳＩＰプロキシが１つしか関与していないが、呼出し元と呼出し先の間のＳＩＰシグナリング経路上には一般に複数のＳ＊ＩＰプロキシがあり、２つ以上のＳＩＰプロキシが、呼出し元クライアントの認証を要求する可能性がある。例えば、図５に示す単純化されたケースでは、ＳＩＰクライアントのアウトバウンドプロキシサーバ７４の他に別のＳＩＰプロキシサーバ１２０があり、両方のプロキシが、ＩＮＶＩＴＥメッセージを転送する前にクライアントの認証を必要とする。この場合には、クライアント７２がまず、図４に関して先に説明した同じプロセスを経て、アウトバウンドＳＩＰサーバ７４に対して自体を認証する。クライアントを認証した後、プロキシサーバ７４は第２のプロキシ１２０にＩＮＶＩＴＥを送り、第２のプロキシ１２０は次いで、クライアントに４０７チャレンジ１９０を送る。これに応答してクライアントは、第２のプロキシサーバ１２０用のケルベロスサーバチケットを含んだＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有する別の新しいＩＮＶＩＴＥ１９２を送る。クライアントを認証した後、第２のプロキシは呼出し先にＩＮＶＩＴＥ１９２を渡す。

【００７４】以下の説明は、ケルベロスまたはＮＴＬＭセキュリティ機構に基づいて認証を実行するさまざまなメッセージフローのシナリオにおいて、Ｐｒｏｘｙ Ａｕｔｈｅｎｔｉｃａｔｅ、Ｐｒｏｘｙ ＡｕｔｈｏｒｉｚａｔｉｏｎおよびＰｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｉｎｆｏｒｍａｔｉｏｎヘッダをどのように使用するかを説明する追加の例を提供する。図６を参照する。このケースでは、ＳＩＰクライアント７２が、体をプロキシサーバに登録するときにケルベロスベースのプレ認証を実行する。クライアントは、プロキシに対するケルベロスサーバチケットを含むＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを含むＲＥＧＩＳＴＥＲリクエスト２００、および先に説明した相互認証のためのリクエストを送る。サーバチケットに基づいてクライアントを認証した後、プロキシは、クライアントが使用してプロキシを認証することができるプロキシ認証データを含むＰｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｉｏｎ－Ｉｎｆｏｒｍａｔｉｏｎヘッダとともに、２００ＯＫメッセージ２０２を返す。ＲＥＧＩＳＴＥＲおよび２００ＯＫメッセージの内容の例を以下に示す。

【００７５】

```
REGISTER sip: nickn@microsoft.com SIP/2.0
    Via; SIP/2.0/UDP www.xxx.yyy.zzz:5060
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markmmarkm@microsoft.com>
    Call-ID: 123456789@microsoft.com
    CSeq: 1 REGISTER
    Contact: <sip:123.45.67.89:5060>
    Proxy-Authorization: Negotiate
realm = "Microsoft RTC Service Provider", qop = "auth", gssapidata = "34
fcbaed78902QWERTY", targetname =
"server1 doaminA.microsoft.com"
```

```
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0
SIP/2.0 200 OK
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
Proxy-Authentication-Info: Negotiate qop = auth, rspauth = "ABCD87564cvx
",
srand = "9876543210" realm = "Microsoft RTC Service Provider" targetname
 =
"serveral.doaminA.microsoft.com"
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com>
    Call-ID: 123456789@ms.com
    CSeq: 1 REGISTER
    Contact: "Nick north" <sip:@www.xxx.yyy.zzz>
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0
```

【００７６】図７に、ケルベロスベースのチャレンジド認証（ｃｈａｌｌｅｎｇｅｄ ａｕｔｈｅｎｔｉｃａｔｉｏｎ）のシナリオを示す。この例では、クライアント７２がまず、Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎ情報を含まないＩＮＶＩＴＥ２０６をプロキシ７４に送る。プロキシはこれに応答して、認証が必要であることを指示するＰｒｏｘｙ－Ａｕｔｈｅｎｔｉｃａｔｅヘッダを含む４０７メッセージ２０８を送る。この４０７メッセージに応答してクライアントは、必要なケルベロス認証データを含むＰｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有するＲＥＧＩＳＴＥＲリクエスト２１０＊を送る。プロキシは、プロキシ自体についての認証情報を含むＰｒｏｘｙ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｉｎｆｏｒｍａｔｉｏｎヘッダを有する「２００ＯＫ」メッセージ２１２を返す。Ｐｒｏｘｙ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｉｎｆｏｒｍａｔｉｏｎヘッダの中のデータに基づいてプロキシを認証した後、クライアントは、Ｐｒｏｘｙ－Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有する第２のＩＮＶＩＴＥ２１４を送る。このプロセスの例示的なメッセージを以下に示す。

【００７７】

```
SIP/2,0 407 Proxy Authorization Required
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com>
    Call-ID: 12345600@PC1.ms.com
    CSeq: 1 INVITE
    Proxy-Authenticate: Negotiate realm = "Microsoft RTC Service Provid
er",
targetriame = "serverl.doaminA.microsoft.com", qop = "auth"
    Contact: <sip:123.45.67.89:5060>
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0

REGISTER sip:nickn@microsoft.com SIP/2.0
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    From: "Nick North"" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com>
    Call-ID: 123456789@microsoft.com>
    CSeq: 1 REGISTER
    Contact: <sip:123.45.67.89:5060>
    Proxy-Authorization: Negotiate realm = "Microsoft RTC Service Provi
der",
opaque = "ABC01234", qop = "auth", gssapi-data = "34fcbaed78902QWERTY"
```

targetname = "server1.domainA.microsoft.com"
User-Agent: Microsoft-RTC/1.0
Content-Length: 0
SIP/2.0 200 OK
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
Proxy-Authentication-Info: Negotiate qop= "auth", rspauth = "ABCD87564cvx", srand = "9876543210"
targetname = "server1.doaminA.microsoft.com" realm = "Microsoft RTC Service
Provider",
From = "Nick North" <sip: nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 123456789@ms.com
CSeq: 1 REGISTER
Contact: <sip:123.45.67.89:5060>
User-Agent: Microsoft-RTC/1.0
Content-Length: 0

【００７８】次に図８を参照する。先に述べたとおり、好ましい実施形態では任意選択で、クライアント・プロキシ認証にＮＴＬＭセキュリティ機構を使用することができる。この場合、クライアントがまず、認証データを含まないＩＮＶＩＴＥメッセージ２２０を送り、プロキシが４０７メッセージを返す。この４０７メッセージ２２０のＰｒｏｘｙ　Ａｕｔｈｅｎｔｉｃａｔｅヘッダは、認証に対してＮＴＬＭを使用しなければならないことを指示する。次いでクライアントが、ＮＴＬＭプロトコルに基づくクライアントの認証データを含むＰｒｏｘｙ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎヘッダを有するＲＥＧＩＳＴＥＲメッセージ２２４を送る。

【００７９】図４の状態機械に関連して先に述べたとおり、クライアントによって送られた認証データによって、プロキシはクライアントを認証することができるが、この認証データに基づいてセキュリティアソシエーションは完全には確立されない。そのためプロキシは、＊Ｐｒｏｘｙ　Ａｕｔｈｅｎｔｉｃａｔｅヘッダをやはり含む別の４０７チャレンジ２２６をクライアントに送る。クライアントは次いで、セキュリティアソシエーションを完成させるのに必要な認証データを含むＰｒｏｘｙ　Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有する別のＲＥＧＩＳＴＥＲリクエスト２２８を送る。プロキシサーバは、第２のＲＥＧＩＳＴＥＲリクエストの中のデータに基づいてセキュリティアソシエーションを完成させ、プロキシについての認証データを含むＰｒｏｘｙ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｉｎｆｏｒｍａｔｉｏｎヘッダを有する「２００ＯＫ」メッセージ２３２を返す。「２００ＯＫ」メッセージ２３２の中の認証データに基づいてクライアントはプロキシを認証し、次いで別のＩＮＶＩＴＥメッセージ２３６を送る。このプロセスの例示的なメッセージを以下に示す。

【００８０】

SIP/2.0 407 Proxy Authorization Required
Via: SIP/2.0/UDP.www.xxx.yyy.zzz:5060
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 12345600@PC1.ms.com
CSeq: 1 INVITE
Proxy-Authenticate: NTLM realm = "Microsoft RTC service Provider",
targetname = "server1.domainA.microsoft.com",
opaque = "ABCD1234", qop = "auth",
Contact: <sip:123.45.67.89:5060>
User-Agent: Microsoft-RTC/1.0
Content-Length: 0
REGISTER sip:nickn@microsoft.com SIP/2.0
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>

```
    Call-ID: 123456789@microsoft.com
    CSeq : 1 REGISTER
    Contact: <sip:123.45.67.89:5060>
    Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",
opaque = "ABCD1234", qop = "auth", gssapi-data = "34fcbaed78902QWERTY"
targetname = "server1.domainA.microsoft.com"
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0
SIP/2.0 407 Proxy Authorization Required
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com>
    Call-ID: 12345600@PC1.ms.com
    CSeq: 1 INVITE
    Proxy-Authenticate: NTLM realm = "Microsoft RTC Service Provider",
targetname = "server1.domainA,microsoft.com", opaque="ABCD1234", qop = "
auth",
    gssapi-data = "QWERTY789564NMJHKLasdcfg"
    Contact: <sip:123.45.67.89:5060>
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0
REGISTER sip:nickn@microsoft.com SIP/2.0
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com>
    Call-ID: 123456789@microsoft.com
    CSeq: 2 REGISTER
    Contact: <sip:123.45.67.89:5060>
    Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",

gssapi-data = "qqertyuioKMNFO09876" opaque = "ABCD1234", qop = "auth",
targetname = "server1.domainA.microsoft.com"
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0
SIP/2.0 200 OK
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    Proxy-Authentication-Info: NTLM realm = "Microsoft RTC Service Prov
ider" qop = "auth",
    rspauth = "ABCD87564cvx", srand = "9876543210"
targetname = "server1.domainA.microsoft.com"
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com>
    Call-ID: 123456789@ms.com
    CSeq: 2 REGISTER
    Contact: <sip:123.45.67.89:5060>
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0
INVITE sip: markm@proxy1.wcom.com SIP/2.0
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    Proxy-.Authorization: NTLM realm = "Microsoft RTC Service Provider"
```

,
crand = "913082051",
response = 12345ABCDEF78909BCADE56", opaque = "ABCD1234", qop =
"auth", targetmane = "server1.domainA.microsoft.com"
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 12345601@PC1.ms.com
CSeq: 2 INVITE
Contact: "Nick North" <sip:nickn@microsoft.com>
User-Agent: Microsoft-RTC/1.0
Content-Type: application/sdp
Content-Length: xxx

【００８１】図９に、ＮＴＬＭベースのプレ認証のシナリオを示す。このケースのメッセージフローは、ケルベロスベースのプレ認証のそれに似ているが、４０７チャレンジおよびＲＥＧＩＳＴＥＲメッセージが追加されている点が異なる。具体的にはクライアントが、ＮＴＬＭを使用することを指示し、ＮＴＬＭ認証データを含むＰｒｏｘｙ Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを含むＲＥＧＩＳＴＥＲメッセージ２４０および相互認証のためのリクエストを送る。プロキシは、受け取ったＮＴＬＭ認証データに基づいてクライアントを認証し、Ｐｒｏｘｙ Ａｕｔｈｅｎｔｉｃａｔｅヘッダを有する４０７チャレンジ２４２を返す。クライアントは次いで、プロキシとのセキュリティアソシエーションを完成するためのＮＴＬＭ認証データを含むＰｒｏｘｙ Ａｕｔｈｏｒｉｚａｔｉｏｎヘッダを有する第２のＲＥＧＩＳＴＥＲリクエスト２４４を送る。プロキシは次いで、Ｐｒｏｘｙ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｉｎｆｏｒｍａｔｉｏｎを有する「２０００Ｋ」メッセージ２４６を返す。プロキシを認証した後、クライアントはプロキシに第２のＩＮＶＩＴＥメッセージ２４８を送る。このプロセスの例示的なメッセージを以下に示す。

【００８２】

REGISTER sip:nickn@microsoft.com SIP/2.0
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 123456789@microsoft.com
CSeq: 1 REGISTER
Contact: <sip:123.45.67.89:5060>
Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",
opaque = "ABCD1234", qop = "auth", gssapi-data = "34fcbaed78902QWERTY",
targetname = "server1.dornainA.microsoft.com"
User-Agent: Microsoft-RTC/1.0
Content-Length: 0
SIP/2.0 407 Proxy Authorization Required
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 12345600@PC1.ms.com
CSeq: 1 INVITE
Proxy-Authenticate: NTLM realrm = "Microsoft RTC Service Provider",
targetname = "server1.domainA.microsoft.com", opaque = "ABCD1234",
qop = "auth",
gssapi-data = "QWERTY789564NMJHKLasdcfg",
targetname= "server1.domainA.microsoft.com"
Contact: <sip:123.45.67.89:5060>
User-Agent: Microsoft-RTC/1.0
Content-Length: 0
REGISTER sip:nickn@microsoft.com SIP/2.0

```
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com.>
    Call-ID: 123456789@microsoft.com
    CSeq: 2 REGISTER
    Contact: <sip:123.45.67.89:5060>
    Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",
gssapi-data = "qqertyuioKMNFO09876" opaque = "ABCD1234", qop = "auth",
targetname = "server1.domainA.microsoft.com"
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0
SIP/2.0 200 OK
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    Proxy-Authentication-Info: NTLM qop = "auth",
  rspauth = "AECD87564cvx", srand = "9876543210",
targetname = "server1.domainA.microsoft.com"
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com>
    Call-ID: 123456789@ms.com
    CSeq: 2 REGISTER
    Contact: <sip:123.45.67.89:5060>
    User-Agent: Microsoft-RTC/1.0
    Content-Length: 0
INVITE sip: markm@proxy1.wcom.com SIP/2.0
    Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
    Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",
crand = "913082051",
response = "12345ABCDEF78909BCADE56", opaque = "ABCD1234", qop "auth" ta
rgetname = "server1.domainA.microsoft.com"
    From: "Nick North" <sip:nickn@microsoft.com>
    To: "Mark Mars" <sip:markm@microsoft.com>
    Call-ID: 12345601@PC1.ms.com
    CSeq: 2 INVITE
    Contact: "Nick North" <sip:nickn@microsoft.com>
    User-Agent: Microsoft-RTC/1.0
    Content-Type: application/sdp
    Content-Length: xxx
```

【００８３】本発明の原理を適用することができる多くの可能な実施形態があることを考慮すれば、図面に関して本明細書に記載した実施形態は例示的なものであって、本発明の範囲を限定するものと解釈してはならないことが認識される。例えば、ソフトウェア（またはハードウェア）として示した記載の実施形態の要素をハードウェア（またはソフトウェア）として実装することができること、または本発明の趣旨から逸脱することなく記載の実施形態の配置および詳細を修正することができることを当業者は認識しよう。したがって本明細書に記載した発明は、前記請求項およびその等価物の範囲に含まれる全ての実施形態を企図する。

【００８４】

【発明の効果】ケルベロスプロトコル、ＮＴＬＭプロトコルなどのセキュリティ機構を、ＳＩＰシグナリングオペレーションのメッセージフローに組み込んで、ＳＩＰクライアントとＳＩＰプロキシが相互に認証できるようにするための方式が提供される。

【図面の簡単な説明】

【図１】本発明を実現することができる例示的なコンピュータシステムを概略的に示すブロック図である。

【図２】セッションシグナリング段階中に相互認証するＳＩＰクライアントとＳＩＰプロキシサーバを含むセッションイニシエーションプロトコル（ＳＩＰ）システムを示す概略図である。

【図３】ＳＩＰクライアントとＳＩＰプロキシサーバの

間の認証目的のシグナリングメッセージ交換を示す概略図である。

【図４】ＳＩＰのフレームワークに組み込まれたセキュリティ機構のオペレーションを表す状態機械を示す概略図である。

【図５】ＳＩＰクライアントが複数のＳＩＰプロキシと認証オペレーションを実行するためのシグナリングメッセージ交換を示す概略図である。

【図６】ケルベロスセキュリティ機構を使用したＳＩＰクライアントとプロキシの間のプレ認証プロセスにおけるメッセージフローを示す概略図である。

【図７】ケルベロスセキュリティ機構を使用したＳＩＰクライアントとプロキシの間のチャレンジド認証プロセスにおけるメッセージフローを示す概略図である。

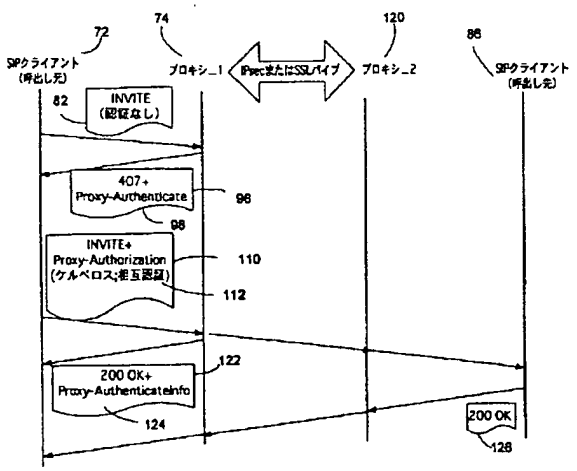
【図８】ＮＴＬＭセキュリティ機構を使用したＳＩＰクライアントとプロキシの間のチャレンジド認証プロセスにおけるメッセージフローを示す概略図である。

【図９】ＮＴＬＭセキュリティ機構を使用したＳＩＰクライアントとプロキシの間のプレ認証プロセスにおけるメッセージフローを示す概略図である。

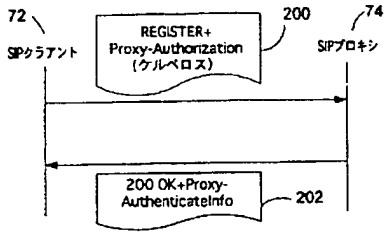
【符号の説明】

２０　パーソナルコンピュータ
２１　処理ユニット
２２　システムメモリ
２３　システムバス
２４　ＲＯＭ
２５　ＲＡＭ
２７　ハードディスクドライブ
２８　磁気ディスクドライブ
３０　光ドライブ
３２　ハードディスクドライブインタフェース
３３　磁気ディスクドライブインタフェース
＊３４　光ディスクドライブインタフェース
３５　オペレーティングシステム
３６　アプリケーションプログラム
３７　その他のプログラムモジュール
３８　プログラムデータ
４０　キーボード
４２　マウス
４６　シリアルポートインタフェース
４７　モニタ
４８　ビデオアダプタ
４９　リモートコンピュータ
５０　メモリ記憶装置
５１　ローカルエリアネットワーク
５２　ワイドエリアネットワーク
５３　ネットワークインタフェース
５４　モデム
７２　ＳＩＰクライアント
７４　ＳＩＰプロキシサーバ
７６、８０　ユーザ
８２　ＩＮＶＩＴＥメッセージ
８６　ＳＩＰクライアント
８８　ワークステーション
９０　シグナリング経路
９６　４０７チャレンジメッセージ
１０２　ドメインコントローラ
１０８　サーバチケット
１１０　ＩＮＶＩＴＥメッセージ
１１６　セッションキー
１２０　ＳＩＰプロキシサーバ
１２２　２００ＯＫメッセージ
１７０、１７２、１７６　ＫＤＣ
＊１８０、１８２　ケルベロスモジュール

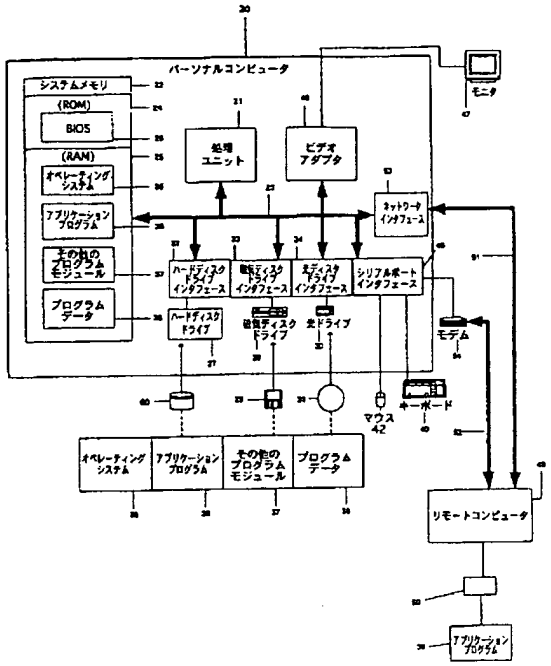
【図３】



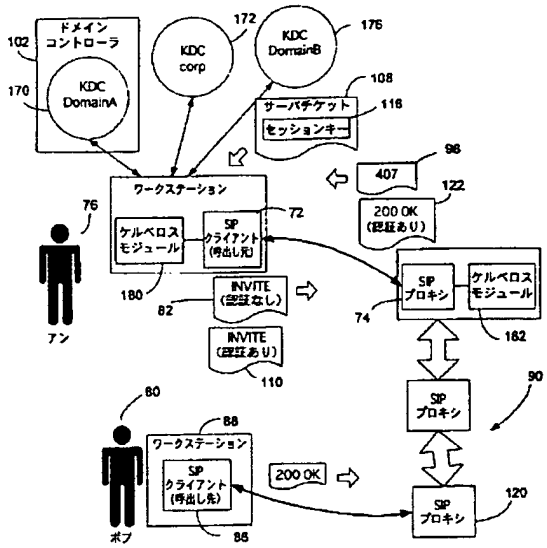
【図６】



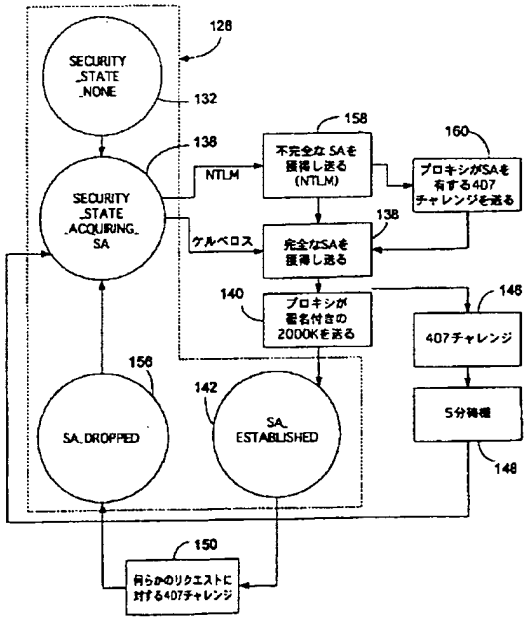
【図１】


パーソナルコンピュータ
システムメモリ
(ROM)
BIOS
(RAM)
オペレーティングシステム
アプリケーションプログラム
その他のプログラムモジュール
プログラムデータ
処理ユニット
ビデオアダプタ
ネットワークインタフェース
ハードディスクドライブインタフェース
磁気ディスクドライブインタフェース
光ディスクドライブインタフェース
シリアルポートインタフェース
ハードディスクドライブ
磁気ディスクドライブ
光ドライブ
モデム
キーボード
マウス
モニタ
リモートコンピュータ
アプリケーションプログラム

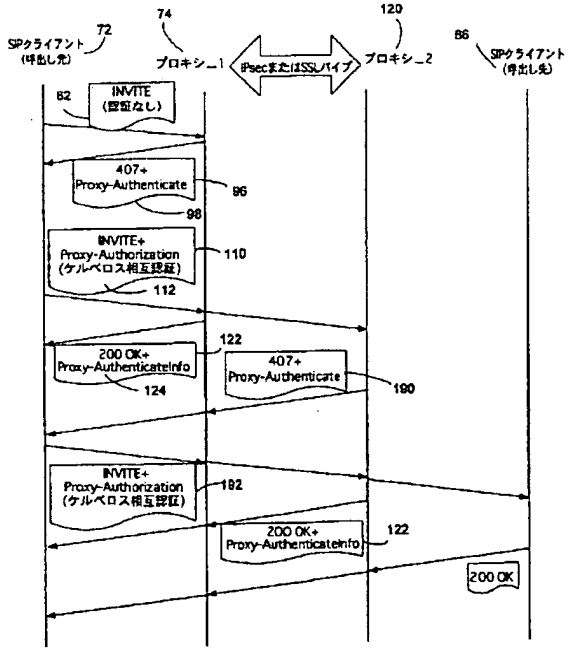
【図２】


ドメインコントローラ
KDC DomainA
KDC corp
KDC DomainB
サーバチケット
セッションキー
ワークステーション
ケルベロスモジュール
SIPクライアント(呼出し元)
407
200 OK(認証あり)
INVITE(認証なし)
INVITE(認証あり)
SIPプロキシ
ケルベロスモジュール
SIPプロキシ
ワークステーション
SIPクライアント(呼出し先)
200 OK
SIPプロキシ
アン
ボブ

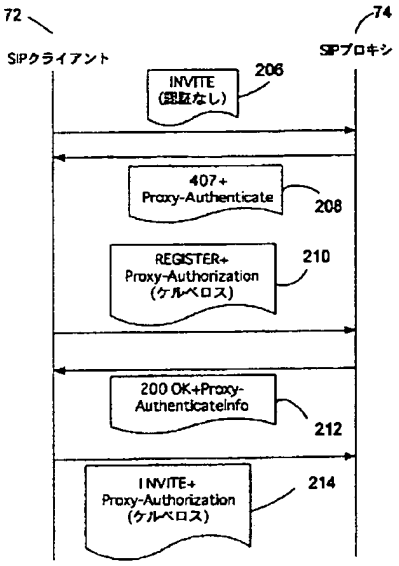
【図４】


SECURITY_STATE_NONE
SECURITY_STATE_ACQUIRING_SA
SA_DROPPED
SA_ESTABLISHED
NTLM
ケルベロス
不完全なSAを獲得し返る(NTLM)
完全なSAを獲得し返る
プロキシがSAを有する407チャレンジを返る
プロキシが署名付きの200OKを返る
407チャレンジ
5分待機
何らかのリクエストに対する407チャレンジ

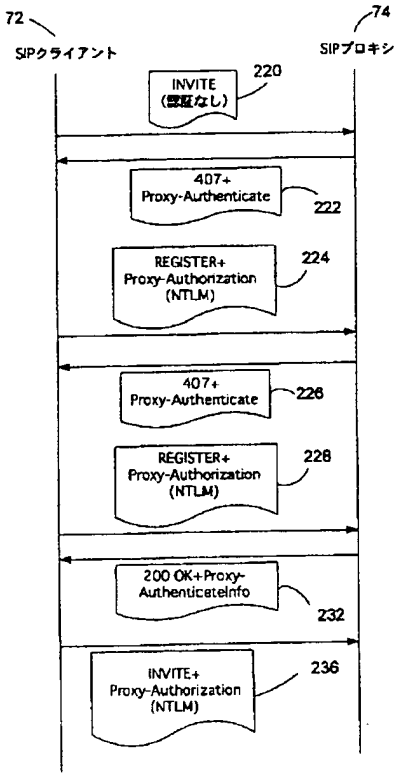
【図５】


SIPクライアント(呼出し元)
プロキシ_1
IPsecまたはSSLパイプ
プロキシ_2
SIPクライアント(呼出し先)
INVITE(認証なし)
407+Proxy-Authenticate
INVITE+Proxy-Authorization(ケルベロス相互認証)
200 OK+Proxy-AuthenticateInfo
407+Proxy-Authenticate
INVITE+Proxy-Authorization(ケルベロス相互認証)
200 OK+Proxy-AuthenticateInfo
200 OK

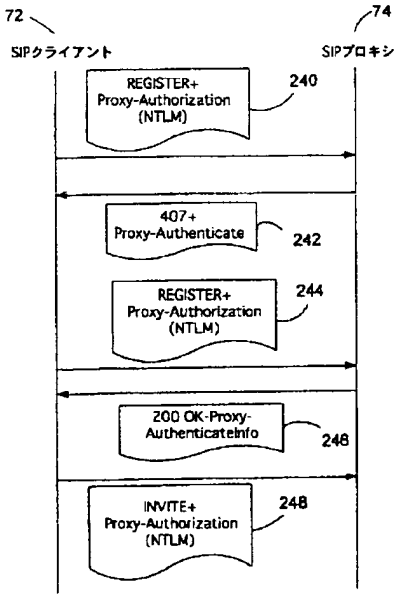
【図７】



【図８】



【図９】



フロントページの続き

(72)発明者 アン デミアジス
アメリカ合衆国 98052 ワシントン州 レッドモンド ノースイースト 67 ストリート 14811

(72)発明者 ムー ハン
アメリカ合衆国 98052 ワシントン州 レッドモンド 153 アベニュー ノースイースト 7204

Ｆターム(参考) 5B085 AE01 BA06 BG02 BG07
5J104 AA07 KA01 PA07